TERRAIN AND SPATIAL EFFECTS ON A HAZARD PREDICTION AND ASSESSMENT CAPABILITY (HPAC) SOFTWARE DOSE-RATE CONTOUR PLOT PREDICTIONS AS COMPARED TO A SAMPLE OF LOCAL FALLOUT DATA FROM TEST DETONATIONS IN THE CONTINENTAL UNITED STATES, 1945-1962

THESIS

Kevin D. Pace, Major, USA

AFIT/GNE/ENP/06-06

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GNE/ENP/06-06

TERRAIN AND SPATIAL EFFECTS ON A HAZARD PREDICTION AND ASSESSMENT CAPABILITY (HPAC) SOFTWARE DOSE-RATE CONTOUR PLOT PREDICTIONS AS COMPARED TO A SAMPLE OF LOCAL FALLOUT DATA FROM TEST DETONATIONS IN THE CONTINENTAL UNITED STATES, 1945-1962

THESIS

Presented to the Faculty

Department of Engineering Physics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science (Nuclear Sciences)

Kevin D. Pace, BS

Major, USA

March 2006

AFIT/GNE/ENP/06-06

TERRAIN AND SPATIAL EFFECTS ON A HAZARD PREDICTION AND ASSESSMENT CAPABILITY (HPAC) SOFTWARE DOSE-RATE CONTOUR PLOT PREDICTIONS AS COMPARED TO A SAMPLE OF LOCAL FALLOUT DATA FROM TEST DETONATIONS IN THE CONTINENTAL UNITED STATES, 1945-1962

Kevin D. Pace, BS
Major, USA

Approved:

Steven T. Fiorino (Chairman) — 3 MAR 06 date

David W. Gerts (Member) — 3 Mar 06 date

Kirk A. Mathews (Member) — 6 Mar 2006

AFIT/GNE/ENP/06-06

Abstract

Hazard Prediction and Capability (HPAC) Software is validated by comparing modeled predictions to historical test data. Reanalysis weather data is acquired and reformatted for use in HPAC. Simulations are made using various amounts of weather data by use of a spatial domain. Simulations are also varied by levels of terrain resolution. The predicted output of the software is numerically compared to historical test data. The result of this research culminated in the knowledge that HPAC prediction accuracy is improved by using terrain resolutions beyond the flat earth assumption. Furthermore, this research establishes that domain size variation produces no significant advantage as to the accuracy of the prediction.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Lt Col Steven Fiorino, for his guidance and support throughout the course of this thesis effort. The patience was certainly appreciated. I would, also, like to thank committee members, Maj David Gerts and Dr. Kirk Mathews for their assistance in this research effort. I would like to give special thanks goes to Dr. Charles Bridgman for taking time to answer a multitude of on-the-spot questions. I also appreciate the assistance received from Dr. Wesley Ebisuzaki and Arthur Taylor of the National Oceanic & Atmospheric Administration and Maj De Leon C. Narcisse of Detachment 3 of the Air Force Weather Agency. I would like to commend Lt. Col Richard W. Chancellor on his ground breaking previous work and willingness and availability to answer a multitude of how-did-you questions.

I am most grateful for the patience and support of my family, without which, this work would have not easily come to fruition. Most importantly, I am indebted to my Heavenly Father for blessing me with the mental capacity to conduct this research. On my own, I am nothing.

Kevin D. Pace

Table of Contents

Page

List of Figures

Figure Page

Figure Page

List of Tables

TERRAIN AND SPATIAL EFFECTS ON A HAZARD PREDICTION AND ASSESSMENT CAPABILITY (HPAC) SOFTWARE DOSE-RATE CONTOUR PLOT PREDICTIONS AS COMPARED TO A SAMPLE OF LOCAL FALLOUT DATA FROM TEST DETONATIONS IN THE CONTINENTAL UNITED STATES, 1945-1962

I. Introduction

Background

The U.S. conducted 1054 nuclear tests [1:59] spanning over 40 years. Tests were conducted for many purposes including proving improved weapon designs and to better understand the phenomenology of nuclear detonations under various physical conditions. From 1945 to 1962, many of these tests were conducted at or near the Earth's surface. These surface detonations created significant amounts of radioactive debris, called fallout. Fallout can pose a serious health risk to local populations.

The modeling of nuclear fallout is a great concern for military leaders and civilian authorities because the effects of fallout on personnel can be devastating. If fallout reaches human skin, it can cause severe burns that take from weeks to months to heal. The long term effects can be just as hazardous in terms of human suffering. Many nuclides contained in fallout have a high rate of uptake in vegetation. Radioactive materials begin causing damage either by direct human uptake or as the nuclides enter the food chain via livestock feeding on this vegetation. These radioactive pathways can lead to a higher rate of cancer in populations by either direct or indirect exposure to fallout materials.

It is imperative that our country has the ability to produce an accurate map of fallout patterns in case of a nuclear detonation on U.S. soil. Though a radiation survey team can accurately map a fallout pattern after a detonation, it is a time consuming process. In order to be responsive immediately after a detonation, prediction models are required. One such model is the Hazard Prediction and Assessment Capability (HPAC) software package produced by the Defense Threat Reduction Agency (DTRA).

Motivation

Modeling a physical phenomenon generally follows the pattern of observing a physical phenomenon, analyzing the physics behind that phenomenon, modeling the phenomenon via simulation, and then using the simulation results as a prediction for a follow-on experiment. This pattern is not possible today in terms of nuclear fallout patterns due to America's self-imposed comprehensive test ban treaty. The inability to test nuclear weapons in the atmosphere causes our modeling process to become truncated as the prediction cannot be used as the expected result of a follow-on nuclear test. Because the U.S. cannot test in the atmosphere, scientists are limited to using historical test data as both our observations and the results of the follow-on experiment. Maps of fallout from historical testing are obtained from "*Compilation of Local Fallout Data from Test Detonations 1945-1962 Extracted from DASA 1251, Volume I – Continental U.S. Tests*" [2] published in 1979 for the Defense Nuclear Agency (DNA). This document is referred to as DASA-EX.

HPAC software cannot be validated by comparing modeled predictions to future physical tests and therefore must be validated by comparing simulation results to historical test data. The predictions made by HPAC are heavily dependent upon meteorological data. However, because HPAC is primarily a prediction model, historical weather data in HPAC is limited. Historical weather data exists but is not readily accessed by HPAC.

Scope

This research focuses on using HPAC software with the most accurate historical weather and terrain data in an effort to model historical nuclear tests involving fallout. This is an extenstion of the work done by LTC Richard W. Chancellor [3] in which he completed a small comparative study between HPAC simulations and historical fallout data. Though HPAC can model other weapon types such as chemical and biological weapons, this work focuses solely on HPAC's ability to model the local residual radiation pattern of a fallout-producing nuclear detonation.

Problem Statement

This thesis addresses three problems. First, there is no publicly available process to transform modern climatological weather data into a form which HPAC can use. I create an automated procedure that produces HPAC weather data files based on the most accurate historical weather data publicly available. Second, HPAC has not been fully evaluated against historical data to determine which, if any, terrain resolution and weather domain is the most accurate. Several simulation runs of HPAC are compared to DASA-EX data using numerical algorithms to identify trends to identify the most accurate method of running HPAC. The final obstacle in HPAC research is the lack of available automated utilities for this program of study. To overcome this obstacle, the major tools and data used in this research are laid out in several appendices that allow future researchers to duplicate and/or modify this procedure.

Approach

The six tests compared in this research were studied by Chancellor [3:3] and are maintained here for continuity. DASA-EX contains test data such as yield of the weapon, height of burst, location of test, and dose-rate contour plots from the fallout. The DASA-EX contour plots are compared with HPAC hazard predictions. This comparison is accomplished using a two-step process that converts the DASA-EX contours into a digital form and then compares this digital information with HPAC results via numerical algorithms. This procedure produces two key metrics - a Cartesian coordinate known as a Measure of Effectiveness (MOE) and a Normalized Absolute Difference (NAD) [4].

HPAC hazard predictions for a given incident are primarily determined by weather and terrain. Chancellor did not use HPAC's terrain feature in his research [3:12]; this study does. Weather data used for building HPAC hazard predictions are available in DASA-EX as single balloon soundings. This is of limited use as fallout patterns are greatly affected by changes in weather, particularly wind speed and direction. Chancellor obtained additional weather data from the Air Force Combat Climatology Center [3:12] which consisted of numerous soundings that were taken at different locations and times in the temporal and spatial vicinity of the test detonations. This weather data allowed HPAC to use four-dimensional weather; 3D space plus time. This weather is not, however, uniform in space or time. The gaps in time and space are not equally spaced and can lead to unnecessary interpolation error. Weather used for this project consists of reanalyzed historical weather data which is a modern weather analysis based on historical weather observations. This data is temporally and spatially gridded. It should be noted that Chancellor did use reanalysis weather obtained through the Air

Force Combat Climatology Center in a follow-on technical report [5]. The conclusions for this follow-on report were based on the same flat-earth assumption used in his thesis. This study culminates with a conclusion as to which parameterization of HPAC best models historical fallout data, if any, and the possible reasons why.

Document Structure:

Chapter 2 summarizes the key physical principles involved in fallout production and how they are modeled in HPAC. A summary of Chancellor's work is also included. Chapter 3 details the method by which this research is to be conducted. It describes how weather reanalysis data is acquired and manipulated for use by HPAC as well as the methodology for how the comparisons were produced. Chapter 4 contains all comparisons of DASA-EX data with the resultant HPAC hazard plots produced as well as an analysis of the results. Chapter 5 summarizes the results, identifies trends, and lists final conclusions about this research. This final chapter also outlines possible reasons why trends appear and attempts to offer explanations as to why one parameterization of HPAC compares better or worse than others. The appendices include information that allows future researchers to reproduce this work. This is important as the reanalyzed weather data used in this research is the first generation of such data. Work is currently under way to produce the second generation of reanalysis weather data.

II. Literature Review

Nuclear Fallout Production

All nuclear detonations in, or reaching, the atmosphere create fallout. The physical characteristics and deposition patterns of fallout is dependent on, but not limited to, several factors including weapon yield, height of burst (HOB), soil composition, and the weather. The following paragraphs provide a brief description of how fallout is produced and transported.

When a nuclear detonation takes place, a fireball with temperatures ranging in the tens of millions of degrees [6: 27] is created. This fireball vaporizes everything it engulfs causing a cloud of gaseous residue to be produced. If a detonation takes place with an altitude exceeding the fallout-free HOB, the only vaporized material contained in the cloud is from the bomb debris itself, which includes the unused fissile fuel, fission fragments, bomb casing, and component material. The fallout-free HOB is found using equation:

$$H \approx 180W^{0.4} \tag{1}$$

where H is the maximum HOB for which there will be appreciable local fallout [6:71]. If the detonation occurs below the fallout-free HOB the fireball reaches the surface and causes soil and other materials present to vaporize. As the fireball grows[1], its temperature decreases, limiting its ability to fully vaporize materials. Therefore, the amount of surface material vaporized is partly determined by how early materials were

[1] The rest of this discussion, and thesis, will on relate to detonations below the fallout-free HOB or shallow underground detonations which produce fallout.

introduced into the fireball's volume. Materials entering the fireball at late times may be introduced as molten particles as opposed to a gaseous state.

When sufficient cooling has occurred, vaporized materials begin to condense into the volume, and onto the surface of the molten particles within the rising fireball. This process continues until all vaporized materials have precipitated out of their gaseous state. At later times, the fireball transforms into a cloud whose color is indicative of the material it contains [6:29]. For detonations over water, the cloud appears white while detonations touching the Earth's surface appear dark. During this time of cooling, the cloud's physical properties of density and temperature allow it to be lofted higher and higher into the atmosphere due to buoyancy.

After only a few minutes have passed, the radioactive cloud will reach a point in space and time where it is in buoyant equilibrium with the surrounding atmosphere. The result is called a stabilized cloud. The height and shape of this cloud are predominantly dependent upon weapon yield and atmospheric conditions [6:31]. For surface bursts, there is also a cloud stem that is created by larger particles whose size and density allow gravity to deny the cloud's updraft created by torroidal motion. Though cloud stabilization is reached quickly, the cloud may grow laterally due to momentum and not buoyancy. Depending on the size of the stabilized cloud, it can take hours for the winds to dissipate the cloud completely. This is dependent on cloud size as well as wind speed and direction. As the cloud continues to cool, the radioactive particles in the cloud begin to settle towards the surface.

Nuclear fallout can be characterized into two categories; early (local) and delayed (global). Local fallout makes up the portion of residual radiation that poses an immediate biological hazard [6:388]. Local fallout only occurs after detonations whose fireball reaches the Earth. Delayed fallout is composed of smaller particles that are lofted to altitudes that allow for a very dilute deposition. Delayed fallout can takes weeks to years to reach the ground and is generally spread over large portions of the Earth's surface. This research only focuses on local fallout – fallout reaching the ground within 24 hours of detonation.

As the radioactive particles are deposited onto the Earth's surface, they decay giving off ionizing radiation as their principal biological damage mechanism. The rate at which radioactivity is absorbed during a time interval is called a dose-rate and is dependent not only upon the amount of radioactivity in the particles in a given area, but also in the way in which that activity is distributed within (or upon) the particles themselves. For example, if radioactive material merges with molten soil in the fireball, then that radioactive material may be volumetrically distributed. This is analogous to dropping a grape into partially solidified gelatin. If, on the other hand, radioactive material is deposited onto a soil particle that has already solidified, the activity of the resulting particle would be surface distributed which is akin to frost deposition on a windshield. The separation of volume- and surface-distributed activity with respect to fallout particles is known as fractionation [7:403]. Fractionation affects the radiation given off by fallout thus affecting measured dose rates. For this study, dose rates are the comparative characteristic as test data only includes this measurable observation.

DASA-EX

The DASA-EX document contains fallout data from various U.S. detonations. Its objective is "to provide a ready reference of fallout patterns and related test data for those engaged in the analysis of fallout effects" [2:2]. The document is organized chronologically by operation and test. For each test the document contains a data sheet which lists dates, times, and locations of the test as well as some basic weather observations. The data sheet also describes the type of burst (e.g. tower, balloon, underground, etc), the HOB, crater data, and any remarks such as how contour dose rates were surveyed and listed. Following the data page, the document contains maps for dose rates both on- and off-site. For this research, only the off-site dose rates were used. Each detonation's section is then concluded with one or more pages of meteorological data.

Table 1 lists the key information for the six tests used in this research. This information was gathered directly from DASA-EX [2]. The location is a key piece of information for this research as terrain is a function of geographical location. It is for this reason that the locations of all selected tests are verified for plausibility using the Google Earth [8] aerial imagery software.

Table 1 Selected Test Data

Test	Date-Time Group	Location (DD.MM.SS)		Yield	Height of Burst
(Operation-Test)	(Zulu)	Latitude (North)	Longitude (West)	(KT)	(ft)
Tumbler Snapper - George	01JUN 52/1155	37.02.53	116.01.16	15	300
Teapot - Ess	23 MAR 55/2030	37.10.06	116.02.38	1	-67
Teapot - Zucchini	15 MAY 55/1200	37.05.41	116.01.26	28	500
Plumbbob - Priscilla	24 JUN 57/1330	36.47.53[2]	115.55.44[2]	37	700
Plumbbob - Smoky	31 AUG 57/1230	37.11.14	116.04.04	44	700
Sunbeam - Johnie Boy	11 JUL 62/1645	37.07.21	116.19.59	0.5	-2

[2] The location of the Priscilla shot is believed to have been incorrectly listed in the DASA-EX document. The DASA-EX document describes the location of the test site as being approximately 125 miles west of the Nevada, Utah, and Arizona border intersection and having an elevation of 3076 ft. Further, the site is named "NTS-Frenchman's Flat". The latitude and longitude were listed as 37.47.53N and 116.55.44W respectively. This location was viewed on Google Earth and shows a location on the side of mountain near Tonopah Test Range Airport which is about 165 miles NNW of the tri-state intersection and has an estimated elevation of 5750 ft. Google Earth revealed an area 19 miles SSE of the NTS with a location of 36.47.53N and 115.55.44W that is named Frenchman Lake with an elevation of 3071 ft. This location is approximately 125 miles west of the tri-state border.

HPAC Methodology

HPAC uses a three step process to model hazardous releases [9:21]. HPAC first

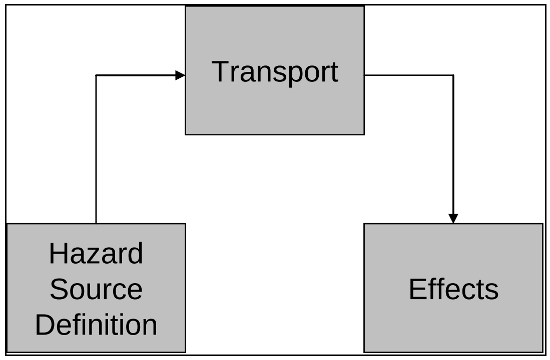


Figure 1 HPAC Transport

creates a hazard source definition from user-supplied input. This definition serves as the foundation of the model by serving as the foundation of the source term (stabilized cloud). In the case of nuclear detonations, this definition consists of weapon yield, HOB, and fission fraction, the latter being the amount of yield produced by nuclear fission as opposed to fusion. This definition is then run through an algorithm in an attempt to model a stabilized nuclear cloud to include height, size, activity, particle-size distribution, and fractionation[3]. This stabilized cloud is then passed to HPAC's transport mechanism which models atmospheric transport and surface deposition based on meteorological data and, if included, terrain. Finally, HPAC will compute dose rates, integrated doses, and even human effects based on the Radiation Induced Performance Decrement algorithm.

HPAC Hazard Definition

Given the same definition for our nuclear hazard source, HPAC defines the stabilized cloud characteristics using an integrated portion of the Defense Land Fallout Interpretive Code (DELFIC). The DELFIC cloud rise model uses observed atmospheric

[3] HPAC uses U238 as the fissile material while true tests may have used other fissile materials.

data and a one- dimensional integration scheme to predict the cloud height at stabilization time. In the absence of DELFIC, cloud height is based on a parameter fit to nuclear test data [9:424]. Aside from the altitude and physical dimensions of the stabilized cloud, there are two other key parameters that must be defined in order to calculate a hazard prediction. They are the particle size distribution and the activity distribution.

To accurately model fallout, a stabilized cloud must be correctly defined in terms of its particle size distribution as well as its activity distribution. The particle size distribution is important as it describes the initial transport position of a group of particles. A small variance in the distribution of initial positions can have large variances in deposition locations due to both horizontal and vertical wind shear. At the same time, the activity distribution within individual particles can also create large disparities between observed and predicted dose rates. As activity distribution within a particle is modeled, it can be shown that incremental ratio changes between surface- and volume-distributed activities markedly change activity readings within fallout areas on the Earth's surface. [7:404-405]

HPAC Weather and Terrain

Weather data must be provided for HPAC to produce a fallout hazard prediction. Weather can be obtained for HPAC in four ways; as a single wind definition, from climatological averages, by downloading weather forecasts from an external server, or via HPAC's built-in weather editor [9:483]. Weather is estimated to be 80% of the solution in an HPAC hazard prediction [10] and is therefore the single most important parameter to characterize correctly when creating decision-making plots for use by military and civil authorities.

Though single wind definitions are the quickest to use, they provide only a single dimension of weather that is copied throughout the spatial and temporal domain of the potential hazard area. This leads to an unnatural linearity in the fallout pattern. Initially, one would think that modeling historical fallout patterns could be accomplished by using the climatological averages. This naïve assumption is quickly dismissed as this climatology data provides weather data that is based on a monthly average and therefore does not represent any single true observation. This data also lacks temporal resolution as it does not change during the time domain of the hazard in which HPAC is modeling. The historical nature of the weather data needed for this work precludes using HPAC's "Weather-Getter" utility which accesses external weather servers. The data that comes from these servers is only stored for a maximum of 30 days [9:622]. HPAC's weather editor allows for the manual building of weather files. For use in modeling historical fallout patterns, the files created by the built-in weather editor are the focus of creating weather definitions for HPAC. A key product of this research is the production of a process that allows the transformation of publicly available historical weather data into a weather file that can be directly used by HPAC.

HPAC has the option of using terrain data when calculating hazard predictions. Terrain data is supplied on the HPAC software DVD in a compressed format. This format contains terrain data that is variable in resolution depending on global location. In general, the resolution varies between hundreds of points per square mile down to only a few. For this research, it was estimated that the Nevada Test Site itself had a resolution of about 9 points per square mile [11]. However, some spatial domains in this research covered a significant portion of the continental United States.

Reanalysis Weather Data [12]

One of the key obstacles in producing ample and accurate weather data for recreating historical fallout patterns is the lack of a centralized database of local observations. Fortunately, the study of historical weather is of paramount importance to climatologists. A climatological project is currently underway to accurately model historical weather in an effort to study climatological changes. This reanalysis project has currently completed a reconstruction of weather data dating back to January 1, 1948.

Reanalysis weather data is constructed by a three-module system. The raw data used for the reanalysis project comes from rawinsonde data, surface marine data, aircraft readings, and satellite measurements to name the key contributors. For the data from 1948 – 1957, however, rawinsonde data is the most available source. This data is gathered and consolidated into a database for use as raw input data for the reanalysis project.

The first module of the reanalysis architecture preprocesses the data to not only check for errors but also data anomalies such as sharp jumps in readings. For example, if two nearby weather stations report drastically different weather readings, the preprocessor can identify this discontinuity and derive a probable solution. This preprocessing ensures that the time-consuming data assimilation module processes the data smoothly without amplifying discontinuities to the point of the algorithm failing.

The T62/28 level NCEP global spectral model is used in the assimilation system, as implemented in the NCEP operational system in December 1994. The model includes parameterizations of all major physical processes, i.e., convection, large scale precipitation, shallow convection, gravity wave drag, radiation with diurnal cycle and interaction with clouds, boundary layer physics, an interactive surface hydrology, and vertical and horizontal diffusion processes. The assimilation model performs an iterative process that ensures the final reanalysis data agrees with the raw input data via spatial and temporal interpolation. The final output of the reanalysis data is a four-dimensional field of weather data.

The reanalysis model's output is formatted in several ways. For this research the output used has a temporal resolution of six hours and a spatial resolution of evenly spaced (latitudinal/longitudinal) points. The latitudinal lines have 180 degrees of coverage represented by 73 points while the 360 degrees of longitude are covered by 144 data points. This translates into a global grid of 73 x 144 points of reanalysis weather data. Each point details weather data at 17 pressure levels that range from 1000 (near surface) to 10 (over 60 km aloft in the standard atmosphere) millibars (mb). The weather data defined at these pressure levels include temperature, height, relative humidity, and wind direction and speed. Though HPAC only requires location, time, wind speed, and wind direction, I am including relative humidity during HPAC modeling under the assumption allowing HPAC to use real weather data alleviates the need to approximate unknown values.

HPAC Transport [9:28-29]

When terrain is used, HPAC calculates a three-dimensional windfield based on the weather data inputs and the specified terrain file. The wind field is determined by first interpolating from the weather data onto a grid and then adjusting the three-dimensional field so that it satisfies mass continuity. A mass-consistent wind model provides a more realistic estimate of the HPAC plume location because the model ensures that air flows around or over (but not through) terrain features. For example, the general wind direction may be deflected by a localized mountain barrier and the wind speed may increase through mountain passes due to the Venturi effect.

HPAC has two integrated mass-consistent wind models, called the Stationary Wind Fit and Turbulence (SWIFT) and the Mass-Consistent Second-order Closure Integrated PUFF (MC-SCIPUFF) model. SWIFT is a more validated model and is used by default. However, SWIFT cannot be used when either the meridional or latitudinal axis of the project domain is 1,000 km or greater or when locations are not entered in via Cartesian coordinates. When SWIFT cannot be used then HPAC automatically switches to MC-SCIPUFF. For this research, only MC-SCIPUFF mode is used as this author was unable to force the consistent use of SWIFT.

HPAC Effects

HPAC has the capability to describe effects in several ways; each of which is based on formulae and/or tabulated data. As DASA-EX only lists dose-rate data normalized to one hour after detonation, this is the effect chosen to be displayed and compared to HPAC simulations in this research. Though HPAC calculates the exact dose rate of any given location within its computational spatial domain and at any given time within its parameterized temporal domain after detonation, the DASA-EX data is limited to dose-rate contours at only a few threshold levels. This work focuses on comparing HPAC output to DASA-EX data using areas enclosed by each contour level listed in DASA-EX.

Measure of Effectiveness

The comparison of contour intervals is accomplished using Warner and Platt's Measure of Effectiveness (MOE) [4:59]. To compute this metric, the figures to be compared are overlapped and categorized into five distinct areas (See Figure 2).

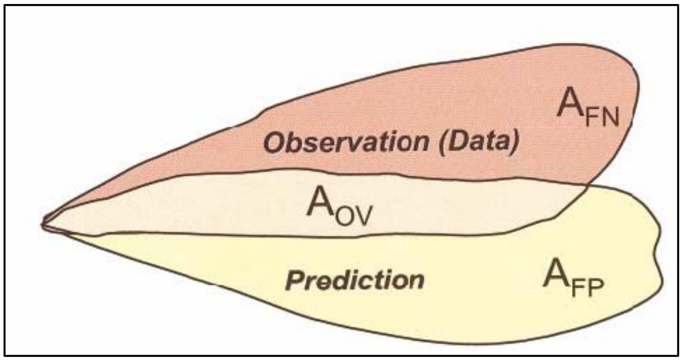


Figure 2 Areal Divisions for MOE Comparison

The first two areas are simply the areas of each individual figure, namely the Area of Observation (AOB) and the Area of Prediction (APR). The other three areas are mutually exclusive. The first is the Area of Overlap (AOV). This is the area where each figure overlaps the other. The areas that are not overlapping are called the Area of False Negative (AFN) and Area of False Positive (AFP). The AFP is the area where the predictive model claims an effect will be made manifest even though the observed data indicates that no such effect was present. The AFN is the conjugate of the AFP in that it describes an area where an effect is known to have occurred but the predictive model failed to identify any such effect at that location.

When the areas are overlapped, in general, there must be at least one point of reference where the areas have commonality. This reference point ensures that, given a perfect model, the areas will be identically shaped and located. For this research there were two reference identities; the release point and relative directions. The release point, or ground zero, for the AOB as well as the APR must be identical in the case of fallout contours in order to ensure equation (2) is satisfied given a perfect model.

$$AOV = APR = AOB \tag{2}$$

If a reference release point were not required, the APR could have the exact same shape as the AOB, but the AOV could be non-existent. Common relative directions are also required. The lack of this trait would have an exactly-perfect model showing contour areas that were identically shaped but in different directions from their release point. Figure 3 illustrates the importance of areal reference properties before performing comparisons. The shape of each area in the chart assumes that the AOB is exactly replicated by a perfect model's APR.

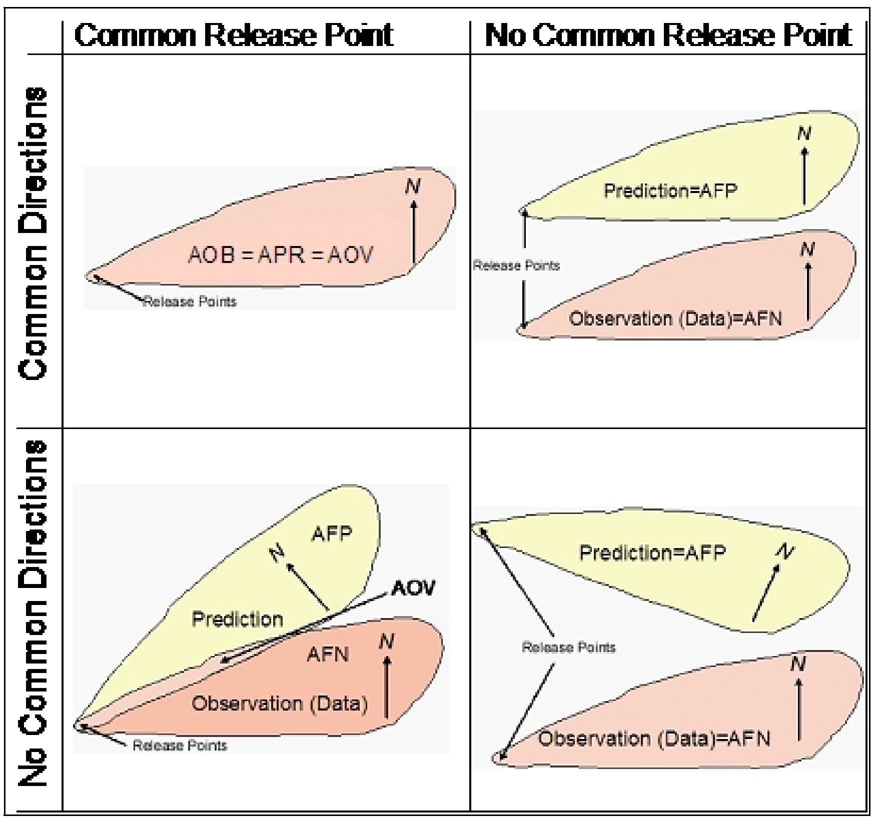


Figure 3 Areal Alignment Chart

When the AOB and APR are aligned correctly, a numerical comparison can be made. The two metrics chosen for analysis are the Measure of Effectiveness (MOE) and the Normalized Absolute Difference (NAD). While the MOE and NAD metrics illustrate how well an HPAC simulation compares to the DASA-EX data, the NAD also allows an objective tool in which to compare HPAC simulation accuracy against each other.

A MOE is a two-dimensional Cartesian coordinate pair whose abscissa (x-coordinate) and ordinate (y-coordinate) are computed according to equation (3):

$$MOE = (x, y) = \left(\frac{AOV}{AOB}, \frac{AOV}{APR}\right) = \left(\frac{AOB - AFN}{AOB}, \frac{APR - AFP}{APR}\right) \qquad (3)$$

This coordinate is plotted on a graph whose x and y axes both range from zero to one. The coordinate near (0,0) is a complete mismatch (absolutely no overlap) and the coordinate of (1,1) is a perfect match (point for point overlap) of the AOB and APR (See Figure 4 Pictorial Representation of MOE Coordinates for a visual explanation of the MOE coordinate system).

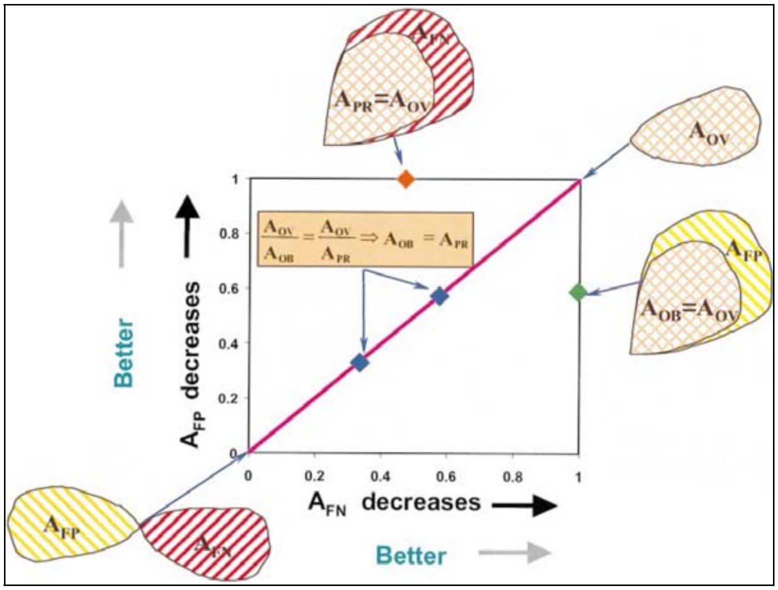


Figure 4 Pictorial Representation of MOE Coordinates

The usefulness of a MOE value resides in not only plotting a single coordinate, but also in simultaneously comparing two model outputs to each other while using the observed data as the standard by which each are measured. By doing this, one can plot MOE locations for the purpose of evaluating which model gives the best prediction to a known standard. Figure 5 illustrates how a MOE can be used to evaluate two models.

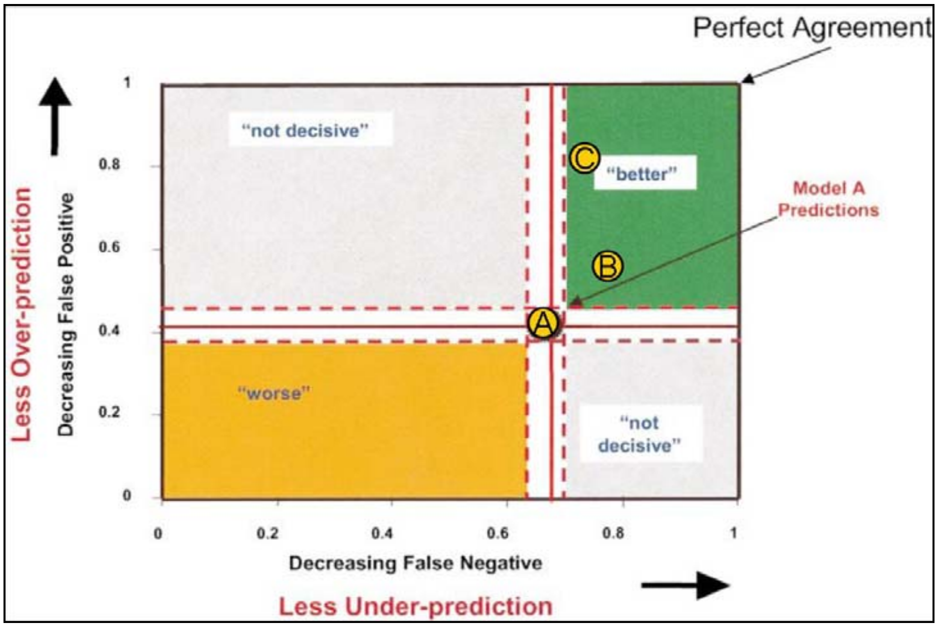


Figure 5 Comparison of MOE Values

This research uses the DASA-EX plots as the known standard and each distinct run of HPAC produces several MOE values; one for each contour (dose rate) level.

Unfortunately, MOE values are of little use when a comparison is required between 3 or more models. For example, if model "A" has a MOE value of (.7, .4) as depicted in Figure 5 and Models "B" and "C" have values of (.8, .5) and (.7, .8) respectively, then it is certain that both "B" and "C" are better than "A". However, when a comparison is made between "B" and "C", it is observed that regardless of which MOE value is plotted first, the other resides in the "not decisive" area. When a comparison cannot strictly determine which model is better via MOE values, another metric is required.

Normalized Absolute Difference

A Normalized Absolute Difference (NAD) [4:65] is a metric that effectively compares models based on how much they deviate from the standard by which they are compared. Figure 5 illustrates that any model whose MOE value resides in the “Better” region is closer to “Perfect Agreement” than model “A” and conversely, any MOE value residing in “Worse” is further from “Perfect Agreement” than model “A”. This indicates that the distance from which a MOE value resides from the coordinate (1, 1) can be used as a standard metric. This idea is integrated into the formation of the NAD.

A NAD represents a normalized value for the distance from the perfect model position of (1, 1). The distance from (1, 1) to any opposing axis is one. For example, in the NAD coordinate system the distance from (1, 1) to (0, .4) is the same (value of 1) as the distance from (1, 1) to (0, 0). Isolines of various NAD values are plotted on the MOE coordinate system (See Figure 6) to further emphasize the relationship between MOE and NAD values.

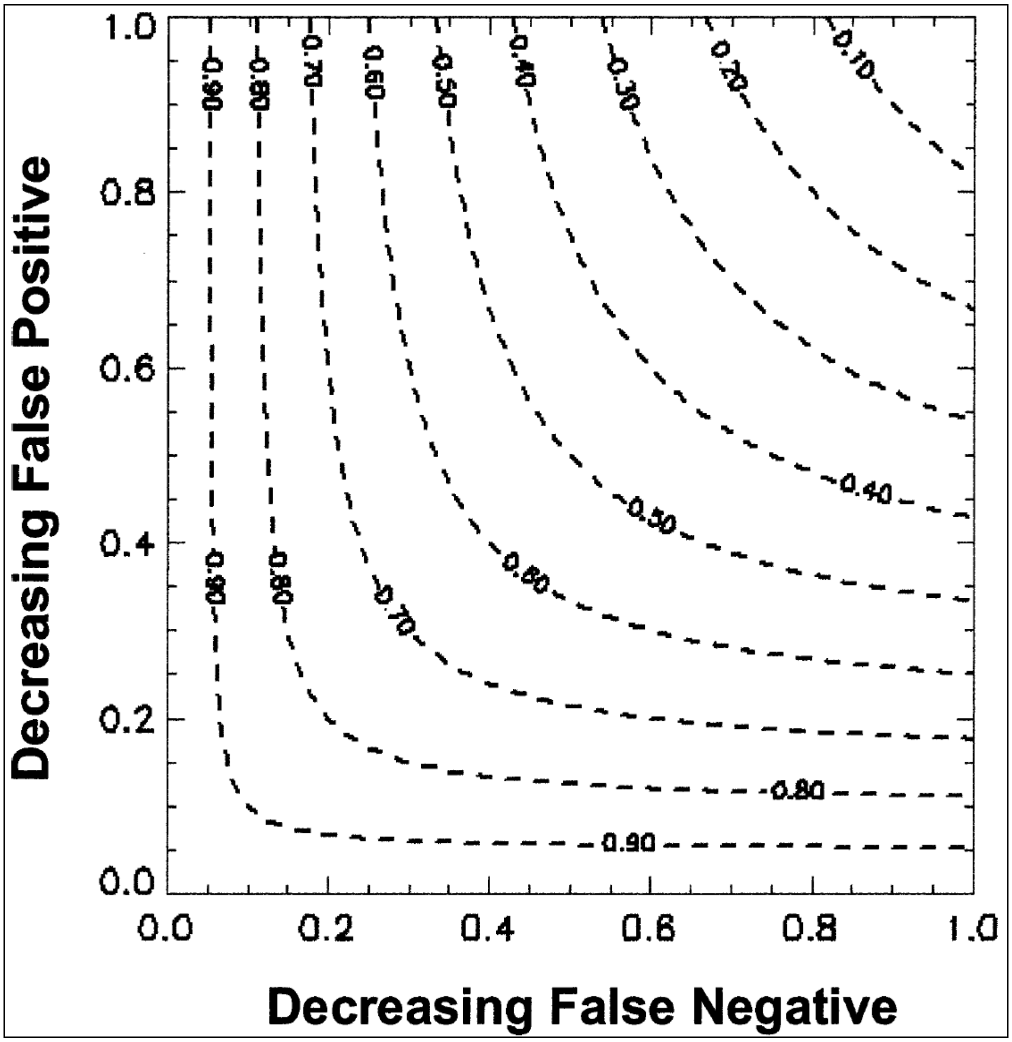


Figure 6 Isolines of Various NAD Values

A NAD takes into consideration a relative weighting factor between AFP and AFN. This relative weighting is useful in making decisions where one factor is much more serious than another. In terms of fallout, it can be argued that large AFPs can unnecessarily waste time, money, and manpower by evacuating populations that are, in actuality, not in danger while large AFNs can result in needless death, illness, and suffering due to an exposed, unwarned population. This research takes the academic approach where neither the AFN nor the AFP has a higher relative importance than the other. Therefore, the AFN and AFP are both relatively weighted at "1".   This results in the following numerical definition of the NAD value:

$$^{4}\, NAD = \frac{AFN + AFP}{2AOV + AFN + AFP} = \frac{x + y - 2xy}{x + y}. \qquad (4)$$

Summary of Previous Research

Chancellor's prediction contours were initially obtained from a run of HPAC using historical Rawinsonde Observations (RAOB). These observations spanning seven days were supplied by the Air Force Combat Climatology Center (AFCCC). One key difference between Chancellor's work and mine is the inclusion of terrain data when creating prediction plots. In his thesis, which compared HPAC versions 4.03 and 4.04, Chancellor concluded that plots produced by version 4.03 compared more favorably than version 4.04 [3:66].

In 2005, Chancellor continued his work using reanalysis weather data supplied by AFCCC [4] in the form of HPAC weather profiles. Contrary to using agency supplied HPAC weather files, my research uses a self-made utility that creates HPAC weather files as part of an internal process. In keeping with his thesis, Chancellor maintained the assumption of a flat Earth in his technical report by excluding the use of terrain data. Chancellor's technical paper revealed that when using reanalysis weather, both versions of HPAC produced similar results with neither version being superior [4].

[4] This equation is mistakenly written as $\frac{y + y - 2xy}{x + y}$ in Warner and Platt's original published paper [4]. Verification is made that equation (4) is correct using Mathematica.

III. – Methodology

Weather Data Manipulation

HPAC acquires weather data by one external and three internal and processes. Internally, HPAC provides two automated weather options; climatology (historical) and single observation (fixed wind) data. These options all carry advantages and disadvantages which must be considered.

Climatology is an archive of historical weather data created by the Air Force Combat Climatology Center (AFCCC) from multi-year records of weather data and is included on the HPAC CD-ROM. This historical weather data includes the effects of both time of day (diurnal) and seasonal variations on the weather. Climatology provides quantitative meteorological input for long-range planning and incidents for which no other weather information is available. [9:591]. Climatological weather data is useful when no weather information can be gleaned for the scenario in which one is interested and/or internet connectivity is not available. However, this data is limited in temporal resolution. Temporally, when you select climatological weather data, the user acquires weather data for a single 24-hour period which repeats for up to 15 days before being updated with a new 24-hour weather definition. These weather definitions consist of weather data for a historically-averaged day that represents a typical day in a given month. For any given location on the Earth, there are only 12 daily definitions available for each year. [9:592]

HPAC also allows for a user to define a fixed-wind field. This essentially fills the spatial domain with an unvarying three-dimensional wind field defined by the user. This weather option is useful when no internet connectivity is available but the basic wind direction of the effected area is known. Due to the lack of variance within the wind field, this option is of use to only those hazards that are limited in temporal duration.

HPAC may also acquire weather data that has been run through modern weather models via the HPAC Meteorological Data Server (MDS). This data consists of weather forecasts for up to five days into the future. These forecasts are kept on the MDS for up to three days. This data is the best available for planning purposes where the user knows the time and location of an HPAC hazard event. Users wanting this data must have an account with DTRA as well as internet connectivity.

Finally, HPAC has the capability to allow users to define their own weather data via an internal weather editor. This editor can be used to modify existing entire weather files or to create entire weather files from scratch. Entering weather into the editor is similar to entering data into a spreadsheet. Allowable data types are shown in Table and are discussed in detail in the HPAC 4.03 user manual [13:7-7-21].

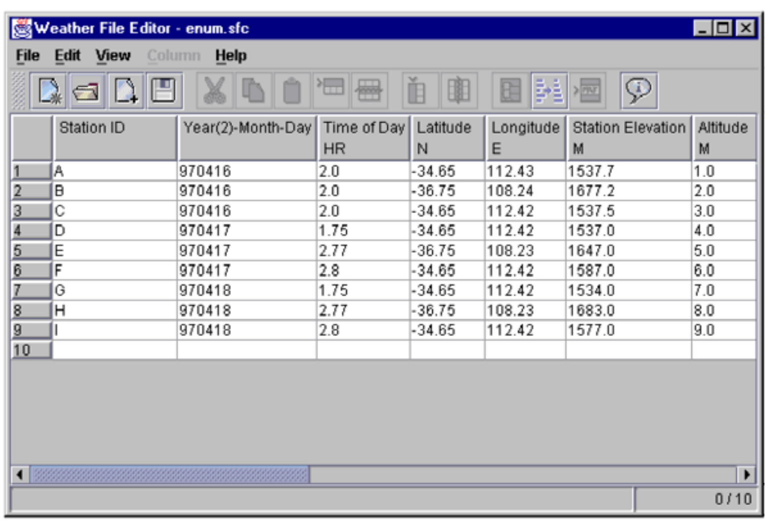
Table 2 HPAC Weather Editor Data Fields

Weather Column	Default Unit
Station ID *	None
Time	Hour
Time of Day *	Hour
Year	None
Month	None
Day	None
Year(2)-Month-Day	None
Year(4)-Month-Day *	None
Julian Day	None
X-Location	km
Y-Location	km
Latitude *	Deg North
Longitude *	Deg East
Station Elevation	m
Mixing Height	m
Stability	None
Sfc. Heat Flux	C-m/s
MO Length	m

Precipitation	None
Wind Speed *	m/s
Wind Direction *	Deg
Altitude *	m
Pressure	mb
Temperature	C
Humidity	%
U-Component	m/s
V-Component	m/s
MU Std. Dev. U	m/s
MU Std. Dev. V	m/s
MU UV Correlation	None
LS Std. Dev. U	m/s
LS Std. Dev. V	m/s
LS UV Correlation	None
Shear Variance	m^2/s^2
Buoyancy Variance	m^2/s^2
Vertical Variance	m^2/s^2
BL Heat Flux	C-m/s
Buoyancy Scale	m
Shear Scale	m

*** Required Weather Data**

Once data has been entered into the weather file, HPAC will manipulate the file from the user-friendly spreadsheet format to a format more suitable for use by the HPAC software. The formats for both user input and software use are shown in Figure 7 and Figure 8.



	Station ID	Year(2)-Month-Day	Time of Day HR	Latitude N	Longitude E	Station Elevation M	Altitude M
1	A	970416	2.0	-34.65	112.43	1537.7	1.0
2	B	970416	2.0	-36.75	108.24	1677.2	2.0
3	C	970416	2.0	-34.65	112.42	1537.5	3.0
4	D	970417	1.75	-34.65	112.42	1537.0	4.0
5	E	970417	2.77	-36.75	108.23	1647.0	5.0
6	F	970417	2.8	-34.65	112.42	1587.0	6.0
7	G	970418	1.75	-34.65	112.42	1534.0	7.0
8	H	970418	2.77	-36.75	108.23	1683.0	8.0
9	I	970418	2.8	-34.65	112.42	1577.0	9.0
10							

Figure 7 Weather Editor User Input Format

```
# CREATOR:        WXEDITOR
# DATE:           2001-04-17 20:58:42 GMT
# SOURCE:         obs
# REFERENCE:      agl
# TYPE:           OBSERVATION
# ANALYSIS:       2001 04 17 12.00
# START:          2001 04 17 12.00
# END:            2001 04 17 15.00
# TIMEREFERENCE:  UTC
# MODE:           profile all
PROFILE
8 6
ID               YYMMDD   HOUR     LAT      LON       ELEV    ZI     HFLUX
                          HOURS    N        E         M       M      W/M2
Z                WDIR     WSPD     P        T         H
M                DEG      M/S      MB       C         %
-9999
ID:  722650      010417   12.00    31.95    -102.22   872     112    -28.68
     2           360      5.1      960      2.6       97
     680         20       19.0     925      3.8       100
     1369        45       14.9     850      4.2       100
     2933        55       12.9     700      -2.9      94
```

Figure 8 HPAC Weather Format for Software Use (Partial View)

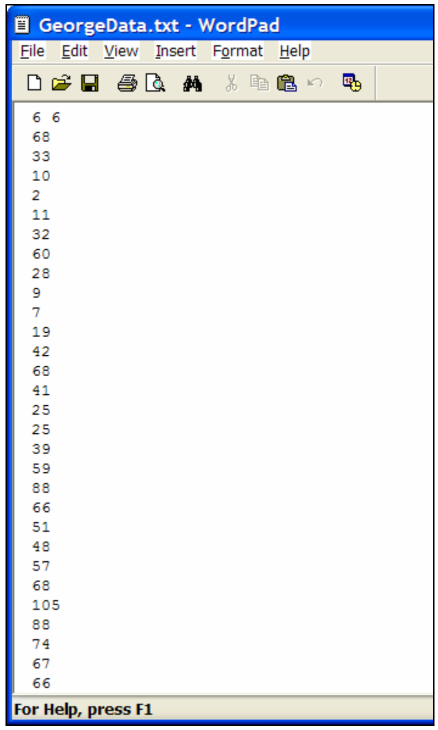
The latter format is discussed extensively in the HPAC 4.03 user's manual [13:7-7-20]. This research takes advantage of the known software format that HPAC uses in calculating hazard predictions.

Transforming reanalysis weather data into the HPAC format is a relatively simple three-step task. Specifically, the process requires the user to download the appropriate weather data, decode the downloaded file with a third-party software utility, and run these files through this author's transformation utility.

Reanalysis weather data is obtained via the National Oceanic and Atmospheric Administration's Operational Model Archive Distribution System (NOMADS) website [14]. Obtaining the weather data consists of downloading a subset of data from the archived weather information as the archive consists of files for each month ranging from January, 1948 through the month prior to the present time. The process, which is described in detail in Appendix A, is marked by a series of 'box checks' to limit size of the file to download. This process begins with selecting the month and year of the data and continues until the weather data is limited to only the weather data used by HPAC covering the desired geographical region.

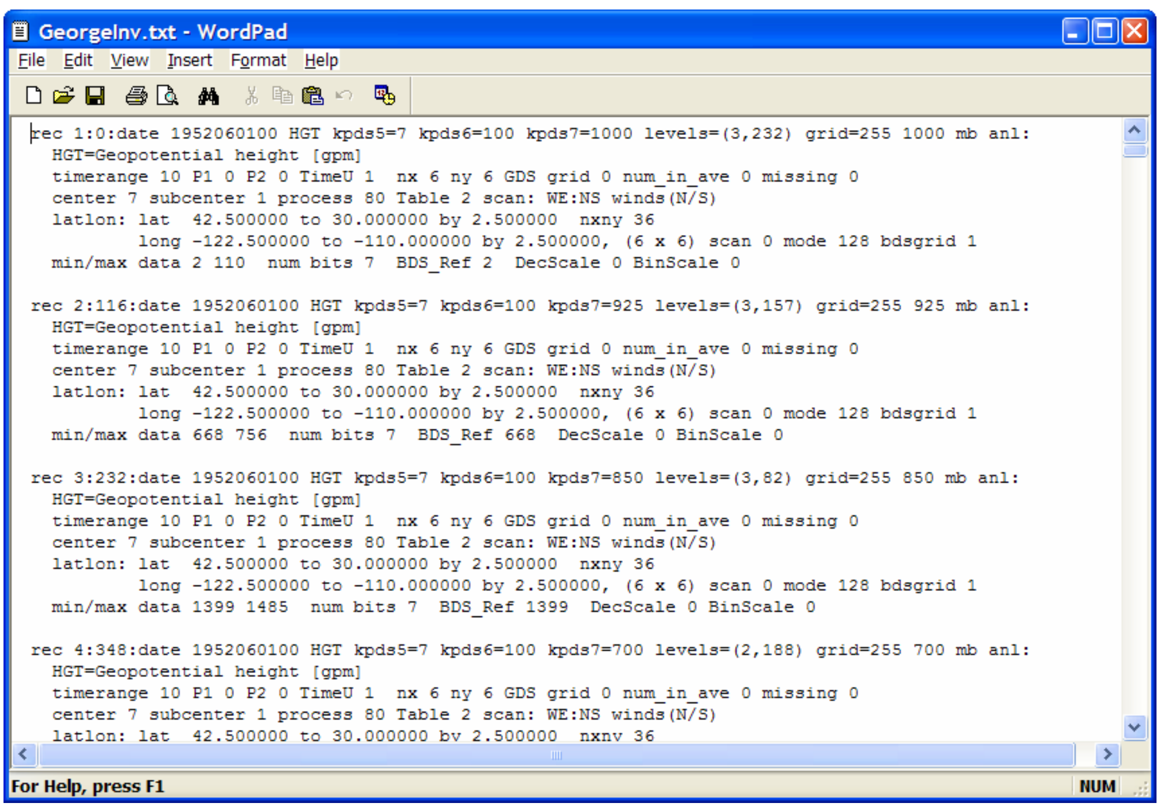
The file obtained by the NOMAD website is known as a “GRiB” file. A GRiB file is a World Meteorological Organization format for GRidded Binary data. GRiB is used by the operational meteorological centers for storage and the exchange of gridded fields. A GRiB file’s major advantage is that its file size is typically 1/2 to 1/3 of the size of normal binary files, the fields are self describing, and GRiB is an open, international standard [15]. The disadvantage of the GRiB file is that non-meteorological sciences cannot read or use the contents of these files without the use of a GRiB-reading software package. This research uses the WGRIB free software utility available on the National Weather Service Climate Prediction Center website.

WGRIB essentially takes a GRiB file as input and operates on it to create two separate files. These files must be used in conjunction with each other as one file contains the raw data (See Figure 9) and the second file contains a description, or inventory, of the data file (See Figure 10). (See Appendix B for WGRIB usage instructions)




```
6 6
68
33
10
2
11
32
60
28
9
7
19
42
68
41
25
25
39
59
88
66
51
48
57
68
105
88
74
67
66
```

Figure 9 Example Data File Decoded by WGRIB Software




```
rec 1:0:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=1000 levels=(3,232) grid=255 1000 mb anl:
  HGT=Geopotential height [gpm]
  timerange 10 P1 0 P2 0 TimeU 1  nx 6 ny 6 GDS grid 0 num_in_ave 0 missing 0
  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
  latlon: lat  42.500000 to 30.000000 by 2.500000  nxny 36
          long -122.500000 to -110.000000 by 2.500000, (6 x 6) scan 0 mode 128 bdsgrid 1
  min/max data 2 110  num bits 7  BDS_Ref 2  DecScale 0 BinScale 0

rec 2:116:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=925 levels=(3,157) grid=255 925 mb anl:
  HGT=Geopotential height [gpm]
  timerange 10 P1 0 P2 0 TimeU 1  nx 6 ny 6 GDS grid 0 num_in_ave 0 missing 0
  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
  latlon: lat  42.500000 to 30.000000 by 2.500000  nxny 36
          long -122.500000 to -110.000000 by 2.500000, (6 x 6) scan 0 mode 128 bdsgrid 1
  min/max data 668 756  num bits 7  BDS_Ref 668  DecScale 0 BinScale 0

rec 3:232:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=850 levels=(3,82) grid=255 850 mb anl:
  HGT=Geopotential height [gpm]
  timerange 10 P1 0 P2 0 TimeU 1  nx 6 ny 6 GDS grid 0 num_in_ave 0 missing 0
  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
  latlon: lat  42.500000 to 30.000000 by 2.500000  nxny 36
          long -122.500000 to -110.000000 by 2.500000, (6 x 6) scan 0 mode 128 bdsgrid 1
  min/max data 1399 1485  num bits 7  BDS_Ref 1399  DecScale 0 BinScale 0

rec 4:348:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=700 levels=(2,188) grid=255 700 mb anl:
  HGT=Geopotential height [gpm]
  timerange 10 P1 0 P2 0 TimeU 1  nx 6 ny 6 GDS grid 0 num_in_ave 0 missing 0
  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
  latlon: lat  42.500000 to 30.000000 by 2.500000  nxny 36
```

Figure 10 Inventory File Decoded by WGRIB Software

The data file's contents can only be understood by using the inventory file's contents. For example, the first 7 lines in Figure 10 describe the meaning of the first section of Figure 9. In this case the data file's first line describes how many grid points worth of data are to be listed, which is 36 (6 points in longitude by 6 points in latitude). The inventory file then describes that the numbers that follow describe the geopotential height for the 1000mb pressure level at each of these data points. The data points, in turn, cover a geographical area that spans from 30 to 42.5 north latitude and 110 through 122.5 west longitude. Also listed is the date and time for which these values are valid.

The values listed in the data file can be read into a software program and manipulated into the HPAC software format. This is done using a FORTRAN [15] utility program that I designed and wrote. Both files are read by FORTRAN where the data values are extracted into a multi-dimensional array. These values are then written to an HPAC compatible file in the proper format. (See Appendix C for the source code of this utility program)

Simulation Variations

The premise of this research is to show what effect, if any, the variations of spatial and terrain resolutions have on the accuracy of HPAC nuclear weapon hazard predictions. This section will focus on not only the specific parameters chosen for simulations, but also the reasons behind them. The simulation parameters chosen can be seen in Figure 11.

Research Simulation Matrix					
		Terrain Resolution (Points per Grid Square)			
		0	900	3500	35000
Spatial Domain Size	Small (Minimum coverage to contain DASA-EX plume distances)	All simulations concluded 48 hours after detonation.			
	Large (30 to 42.5N Latitude and 110 to 122.5W Longitude)				

Figure 11 Research Simulation Matrix

The spatial domain on any HPAC simulation is varied according to need. Domain Size in this research is listed as the number of reanalysis grid points used. For example, 3x4 represents an area whose east-west width is defined by three grid points (5 degrees longitude) and whose north-south length is defined by four grid points (7.5 degrees of latitude) for a total of 12 grid points. The full spatial grid of data points for this research is illustrated in Figure 12. Note that the grid points are numbered in the order that the decoded GRiB data file lists them. Initially, it was believed that the amount of weather data used in an HPAC simulation would make a difference in the accuracy of the prediction. The Nevada Test Site is located within the grid with corners numbered 15, 16, 21, and 22. Just as modern interpolation routines generally get better with more surrounding data points, it was believed that additional 'layers' surrounding the detonation site would provide for a more accurate interpolation of the weather. This, in turn, would provide a prediction which was more accurate. This was tested by choosing two distinctly sized spatial domains in which to compare. As opposed to the full 6x6 spatial domain (large), a minimum domain (small) contained only enough data points to contain the fallout plume for a distance at least as large as the observed data.

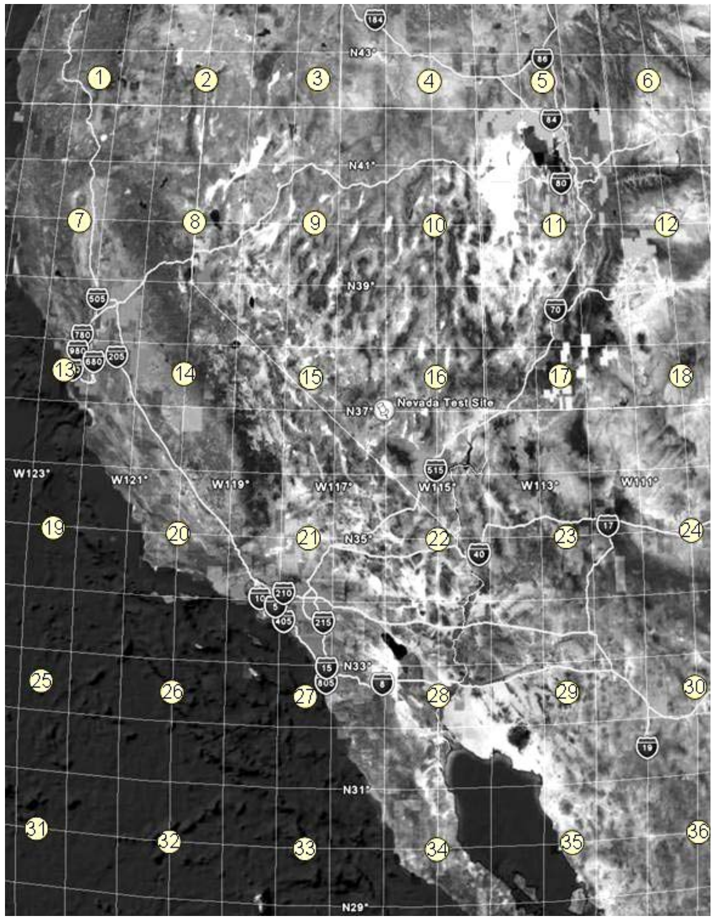


Figure12 Large Spatial Domain Grid Pattern for Reanalysis Weather

For example, if a DASA-EX fallout plume was recorded out to a distance 160 miles east of ground zero and then cut off, then the only data points beyond 15, 16, 21, and 22 might be 17 and 23. This is the way in which the “small” domain was calculated. Even though the fallout plume might extend further in HPAC, there is no reason to track this hazard area as there is no observation in which to compare it. On the other hand, if the observation data is not cut off, the entire HPAC plume is used in the comparison.

Other spatial domain sizes were considered. A series of simulations were run using the data from the George shot. When comparisons were made from full to minimum spatial domain, it was clear that the greatest difference lay at the ends of the size spectrum. The second consideration in choosing domain sizes was time allocation.

Only two domain sizes were chosen as the time allotted for this research was limited. Thus the two domain sizes chosen, large and small, were done so as a way to best show a concurrence or nonoccurrence with the hypothesis that more weather data included in a simulation equates to a more accurate nuclear hazard prediction by HPAC.

Past research considered weather day that began on D-Day and ended at D+7. HPAC simulations range from 20 seconds to over two hours of processing time. This processing time is directly proportional to the temporal domain and level of terrain resolution of the simulation. Early in the research, a short interrogation of these two factors was completed. The purpose was to, given a limited computing time budget, determine if it were more important to spend computational time due to higher terrain resolution or a longer temporal domain. Several simulations were conducted in which the terrain resolution was held constant and the temporal domain was varied and vice versa. The outcome was clear that differences in nuclear hazard predictions varied significantly more with higher terrain resolution than with a longer temporal domain. In fact, in keeping with the fact that most local fallout deposits with 24 hours [6:37], HPAC's output varies little as a function of time for times greater than 24 hours. However, investigation revealed that there is even less variation after 48 hours. Given these facts, a decision was made to limit the temporal domain to 48 hours post-detonation in order to maximize the terrain resolution effect.

Terrain resolution in HPAC is defined by the amount of gridding done on the active spatial domain. That is to say, given a spatial domain, a terrain resolution of 900 equally divides the geographic area into a grid of 900 rectangles. Roughly, the latitudinal and longitudinal components of the spatial domain are divided into the square root of the terrain resolution. In the case of a 900 point terrain resolution, there are approximately 30 latitudinal divisions and 30 longitudinal divisions. Terrain resolutions are variable with values ranging from about 30 to "Native" which is the raw terrain data from the HPAC DVD. Native resolution contains resolutions that range from 9 points per square mile for some uninhabited areas all the way to urban areas where the terrain is described by hundreds of points per square mile. Typically, the higher the terrain resolution, the larger the terrain file that is created. During the processing study, it was found that for HPAC running on a high-end workstation, there is a practical limit to the terrain resolution that can be used in a simulation. After a terrain resolution of about 35,000 points, HPAC is not able to produce results (system crashes) or is not able to do so with any practical efficiency (test runs taking over two hours). The terrain resolutions chosen for this research were 0 (no terrain; flat Earth assumption), 900, 3500, and 35,000 (35K) point.[5] Chancellor ran all his tests using no terrain and this research's objective is to document whether terrain has an appreciable effect on the accuracy of the prediction, and if so, by how much.

In order to have like comparisons, simulations for no terrain are to be executed using the same spatial and temporal domains as the simulations which used terrain. It is found that the difference in hazard prediction between the previous and current research are noticeable but not drastic. However, it cannot be concluded that temporal and spatial domain differences, nor the difference in versions of HPAC, are the cause for these

[5] For all tests, the highest terrain resolution in a scenario is .75 points per square mile. This relatively low resolution prevents unnecessary error by interpolating beyond the native resolution.

differences as there are some fundamental differences in which the research was conducted. These differences are pointed out as appropriate.

HPAC Settings

HPAC simulations are customized by manipulating one or more of the many settings contained within the software. Besides the obvious choices that must be made such as location, time, and event characterization, there are several modeling factors that can be varied from their default settings depending upon the needs and knowledge of the user. For continuity, the settings for this research's simulations are listed in Appendix D.

Numerical Comparisons

Numerically comparing the HPAC simulation with historic test data is a three-step process which includes digitizing the historic observations, running HPAC simulations, and then manipulating each set of data for comparison via a numerical algorithm. This section discusses, in moderate detail, each step of the process. For more detail and the source code for last step in this process, see Appendix E.

The first obstacle in executing a numerical comparison is the conversion of DASA-EX observation data from hardcopy to an appropriate electronic version. The procedure takes advantage of the step-wise representation of the DASA-EX contour data. This terracing of dose rates is easily rendered into a digital image using Canvas software [17]. The goal of this procedure is to re-represent geographical areas from scaled areas on paper to individual representative pixels in a digital image. While each square mile is represented by a scaled area of paper on the hardcopy DASA-EX document, each square mile is represented by 9 pixels (3 pixels per linear mile) in the digital format. This pixel scaling is in keeping with past research and has practical value when exporting HPAC simulation data. Further, each digital image is oriented with north at the top of the image and east to the right of the image. This allows for a less cumbersome comparison algorithm. The process of digitizing DASA-EX data takes geographical areas and dose-rate values into consideration.

The digital image created is done so in grayscale. This allows for the assignment of a scalar value (0-255) to each pixel. This value represents the dose rate depicted by DASA-EX. This grayscale value serves two purposes. The first, and most important, reason for the grayscale format is the ease in which computerized comparisons can be accomplished. Grayscale pixels have a single numerical value which represents its darkness while other formats, such as RGB, have a triplet of values for each pixel. While pure red, green, and blue pixels can easily be used for a numerical comparison, colors that use a combination of these colors cause the comparison process to become convoluted. Additionally, the casual observer is best suited to see that darker gray equates to higher dose rates while color assignment based on dose rate is somewhat arbitrary. Second, grayscale formatting allows for unencumbered viewing of the digital image in black and white publications. Once this digital image is constructed, the Canvas software has the ability to export the image as a table of values; essentially a representative two-dimensional matrix where each value represents an individual pixel's grayscale value (See Figure 13). For this research, the lowest dose rate for a given test shot is represented by a grayscale value of 225 (very light gray) and higher dose rates

130	130	130	130	130	130	130	130	130	130	130	146	175	175	175	175	175	175	183	200	200
130	130	130	130	130	130	130	130	130	130	130	149	175	175	175	175	175	175	185	200	200
130	130	130	130	130	130	130	130	130	130	130	151	175	175	175	175	175	175	185	200	200
130	130	130	130	130	130	130	130	130	130	130	153	175	175	175	175	175	175	185	200	200
130	130	130	130	130	130	130	130	130	130	130	156	175	175	175	175	175	175	185	200	200
130	130	130	130	130	130	130	130	130	130	130	156	175	175	175	175	175	175	185	200	200
130	130	130	130	130	130	130	130	130	130	130	157	175	175	175	175	175	175	186	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	186	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	186	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	186	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	186	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	187	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	188	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	188	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	188	200	200
130	130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	188	200	200
130	130	130	130	130	130	130	130	130	130	130	161	175	175	175	175	175	175	189	200	200
130	130	130	130	130	130	130	130	130	130	130	161	175	175	175	175	175	175	190	200	200
130	130	130	130	130	130	130	130	130	130	130	161	175	175	175	175	175	175	190	200	200
130	130	130	130	130	130	130	130	130	130	130	161	175	175	175	175	175	175	190	200	200
130	130	130	130	130	130	130	130	130	130	130	163	175	175	175	175	175	175	190	200	200
130	130	130	130	130	130	130	130	130	130	130	166	175	175	175	175	175	175	191	200	200
130	130	130	130	130	130	130	130	130	130	130	169	175	175	175	175	175	175	191	200	200
130	130	130	130	130	130	130	130	130	130	130	175	175	175	175	175	175	175	192	200	200
130	130	130	130	130	130	130	130	130	130	130	175	175	175	175	175	175	175	193	200	200
130	130	130	130	130	130	130	130	130	130	133	175	175	175	175	175	175	175	193	200	200
130	130	130	130	130	130	130	130	130	130	136	175	175	175	175	175	175	175	194	200	200
130	130	130	130	130	130	130	130	130	130	139	175	175	175	175	175	175	175	194	200	200
130	130	130	130	130	130	130	130	130	130	143	175	175	175	175	175	175	175	195	200	200
130	130	130	130	130	130	130	130	130	130	146	175	175	175	175	175	175	175	196	200	200
130	130	130	130	130	130	130	130	130	130	149	175	175	175	175	175	175	175	196	200	200
130	130	130	130	130	130	130	130	130	130	152	175	175	175	175	175	175	175	197	200	200
130	130	130	130	130	130	130	130	130	130	156	175	175	175	175	175	175	175	197	200	200
130	130	130	130	130	130	130	130	130	130	159	175	175	175	175	175	175	175	197	200	200
130	130	130	130	130	130	130	130	130	130	162	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	130	165	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	130	169	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	130	172	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	130	175	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	130	175	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	132	175	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	133	175	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	136	175	175	175	175	175	175	175	175	200	200	200
130	130	130	130	130	130	130	130	130	137	175	175	175	175	175	175	175	177	200	200	200

Figure 13 Canvas Exported Matrix of Values (Partial View Only)

are represented by successively lower (darker) values. Specific values are chosen based on the ability to visually discern grayscale values.

HPAC software allows analysis of the hazard prediction visually and numerically. Visually, hazard predictions are viewed in a myriad of ways including dose-rate, integrated dose and casualty estimate contours. Numerically, dose rates and integrated dose rates are exported to a text file. In contrast to past research, visual hazard predictions are not converted into digital images, rather, HPAC dose-rate values are directly used for the comparison. This reduces error introduced in the digitization process including scaling, orientation, and contour-tracing errors.

The exported numerical dose-rate file contains several lines of header data followed by a sequential list of values at specified locations (See Figure 14). The locations listed are based on user input. HPAC requires the user to set at least two points to define a rectangular geographical area. The two points used are the southwestern and

```
#Model ModelName    :HPAC
#Model ModelVersion:4.04.058
#Model Description :Atmospheric dispersion hazard prediction model
#Model Comments     :Sponsor:Defense Threat Reduction Agency (DTRA)
#Model Comments     :Model   :Hazard Prediction and Assessment Capability
#Output CreationDate   :Mon Dec 12 14:44:43 EST 2005
#Output Description    :U238TN(Dose Rate);NWPN Radiation Dose;03-Jun-52 12:00:00Z (2.500004 day)
#Output Source         :HPAC project: client=George 6x6 48hr No Terrain server=George 6x6 48hr No Terrain
#Output Classification:Unclassified
#Output Analyst        :Unknown
#Output Comments       :Project date    :Mon Dec 12 14:36:26 2005
#Output Comments       :Project version:4.04.058 | T:4.04.012-S:2.2
#Output Comments       :Project title  :Unknown
#Export :ModelOutputPointData (simple ASCII)
#Time   :06/03/1952 at 12:00:00
#Notes  :U238TN(Dose Rate)
#Notes  :NWPN Radiation Dose
#Notes  :03-Jun-52 12:00:00Z (2.500004 day)
#Tag    :Model            :HPAC;NWPN Special Edition
#Tag    :SouthWest Corner:(-122.50000E, 30.00000N)
#Tag    :NorthEast Corner:(-110.00000E, 42.50000N)
#Tag    :Field min value :1.0E-30 Rad/hr
#Tag    :Field max value :4.3736973 Rad/hr
#Field 1           :Mean
#Field 1 Units     :Rad/hr
#Field 1 Max Value :4.3736973
#Field 1 Min Value :1.0E-30
G0        (-116.92781E, 36.75859N)    2.869923E-13
G1        (-116.92174E, 36.75859N)    2.870592E-13
G2        (-116.91567E, 36.75859N)    2.871261E-13
G3        (-116.90961E, 36.75859N)    2.871931E-13
G4        (-116.90354E, 36.75859N)    2.872601E-13
G5        (-116.89747E, 36.75859N)    2.873270E-13
G6        (-116.89140E, 36.75859N)    2.873940E-13
G7        (-116.88533E, 36.75859N)    2.874610E-13
G8        (-116.87926E, 36.75859N)    2.875280E-13
G9        (-116.87319E, 36.75859N)    2.875951E-13
G10       (-116.86713E, 36.75859N)    2.876621E-13
G11       (-116.86106E, 36.75859N)    2.877292E-13
G12       (-116.85499E, 36.75859N)    2.871460E-13
G13       (-116.84892E, 36.75859N)    2.791261E-13
G14       (-116.84285E, 36.75859N)    2.713204E-13
G15       (-116.83678E, 36.75859N)    2.637425E-13
G16       (-116.83072E, 36.75859N)    2.563762E-13
```

Figure 14 Exported HPAC Dose-Rate Data (Partial View Only)

northeastern corners of the area of interest. This geographical area is then divided up into a user-specified number of points. HPAC requires the user to supply the number of points in which the defined area is to be divided on both the x- (north-south) and y- (east-west) axes. HPAC is limited to 1000 divisions on either axis. It is convenient that the scale of three pixels to one linear mile is used as the largest area of consideration for this research was 270 by 100 miles (810 by 300 points) as described by the Smoky Test.

A numerical comparison can now be made as both the observation and prediction data are in numerical form, albeit in slightly different formats. The approach for the comparison algorithm is to fill two identically shaped two-dimensional arrays with the test and prediction data. The arrays are formatted such that each location in the respective array represents a geographical location for that data relative to ground zero (ground zero is located at the center of the array). Furthermore, the value contained in that location corresponds to the dose rate at that geographical location. Moreover, the location in one array corresponds to the same location in the second array. That is to say, the top, right array location in the observation data represents a specific point on the ground and its value represents the observed dose rate at that location. At the same time, the top, right array location in the prediction data represents the same geographic location as the observation data but its value represents the predicted dose rate. These matching formats allow a systematic point-to-point comparison between the observed and predicted hazard areas.

It is important to ensure that both sets of data represent like conditions. The DASA-EX document states that "the dose-rate contours for the fallout patterns have been drawn to show the gamma dose rate in roentgens per hours [*sic*], three feet above the ground, in terms of the one hour after burst reference time. The $t^{-1.2}$ approximation was used…" [1:2]. All of these factors were accounted for in this research.

HPAC dose rates for fallout are a function of the gamma and beta activity at a certain time after detonation, and at a height above the ground [9:638]. Even though HPAC takes beta activity into consideration, its effect on observed data is negated for all beta particles whose energy resides below about 3 MeV [18:129] due to the fact that dose-rate measurements were taken at three feet above the Earth's surface. In fact, for fission products resulting from a nuclear detonation, the overall beta-particle spectrum is dominated by energies of less than 1 MeV [19:30].

HPAC dose-rates have units of Rad/hr whereas the DASA-EX data is in roentgens/hr. The HPAC 4.04 User's Manual states [9:649]:

$$1 \text{ REM} = 1 \text{ RAD} \tag{5}$$

and the HPAC 4.03 User's Manual states the conversion as [13:H-6]:

$$1 \text{ REM} = 0.7 \text{ Roentgen}^{6} \tag{6}$$

Normalizing dose-rate values to one hour after detonation is accomplished using HPAC's radioactive decay power law [13:H-6]:

$$R(t) = R_0\left(\frac{t - t_{rel}}{t_0}\right)^{-p} \tag{7}$$

where

R = dose rate,

R_0 = reference dose rate

t_0 = reference time (1 hour)

t_{rel} = time of release

p = decay power = 1.3[7]

These conversion factors were used in the comparison algorithm. The specifics of this methodology can be seen in the FORTRAN source code contained in Appendix E.

[6] Previous research used no conversion factor between REM and Roentgen values

[7] Previous research used a decay power value of 1.2 IAW the Way-Wigner approximation [6:426]. This difference causes my research to re-run HPAC simulations using no terrain in order to make like comparisons.

IV. Results and Analysis

Chapter Overview

This chapter is to presents visual and numerical comparisons resulting from the previous chapter's methodology. The chapter will initially treat each test separately and then group tests according to similar attributes in order to gain as much information as possible. The individual test sections begin with visual observations of the DASA-EX data. This is followed by remarks of how well HPAC simulations compared to the DASA-EX data. There is also, if appropriate, visual comparisons made between HPAC simulations. Visual observations and comparisons are then numerically represented using MOE and NAD metrics.

Operation Tumbler Snapper – George

The DASA-EX document illustrates six dose-rate contours. These contours are listed as "off-site". The area in which contour data was taken extends approximately 200 miles downwind of ground zero. Figure 15 displays a northerly-oriented digitized representation of the DASA-EX contour plot.

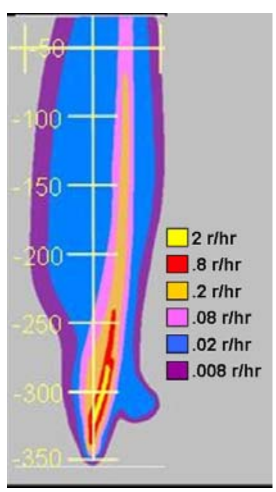


Figure 15 George Digitized Image from DASA

The key feature of this plot is the large protrusion of the .02 and .008 roentgen/hr contour lines located approximately 60 km northeast of ground zero (ground zero located at the intersection of the "-350" tick mark and the y-axis). Research into this feature reveals that the area of the protrusion is a mountain ridge. Figure 16 is an image exported from Google Earth that shows the ground zero mark of the George shot overlaid on the surrounding terrain. The arrow from ground zero to the ridge of the mountain is

Figure 16 Detonation Terrain (George)

approximately 60 km. From this image, it is apparent that terrain does play an important and real role in the deposition location of fallout.

HPAC simulation output is depicted in Figure 17 and Figure 18. These images are all northerly oriented and show by which parameter the simulation was computed; terrain resolution and spatial size. The gray squares in the background of Figure 18

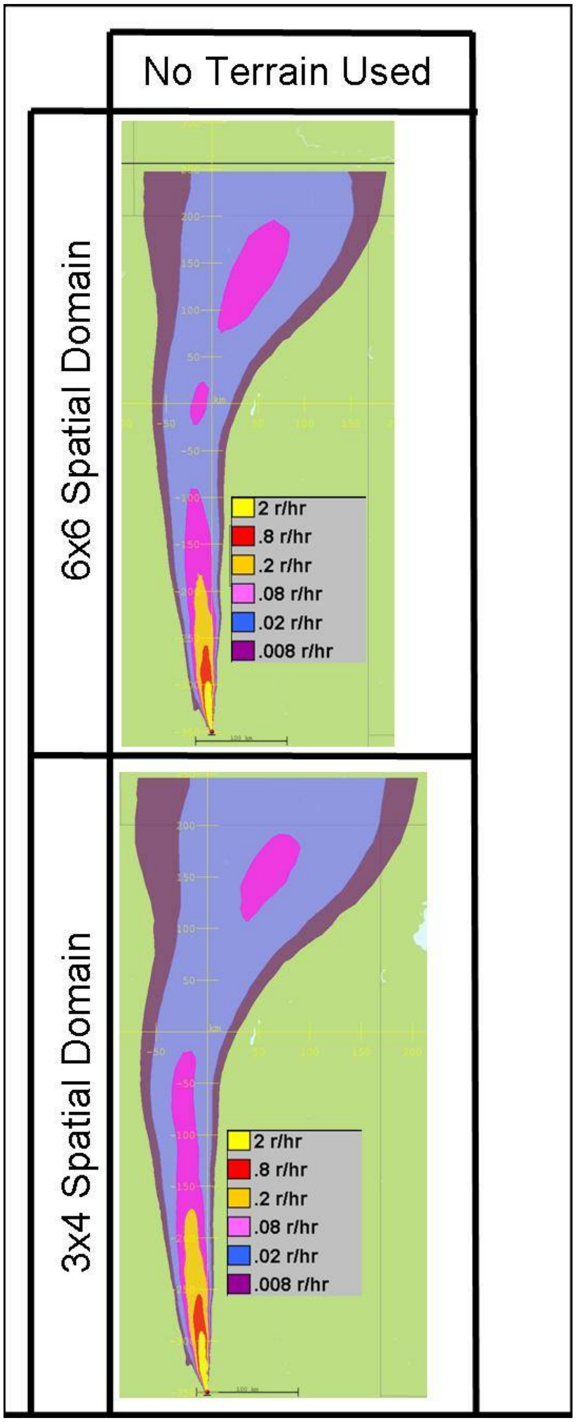


Figure 17 George Simulations Using No Terrain

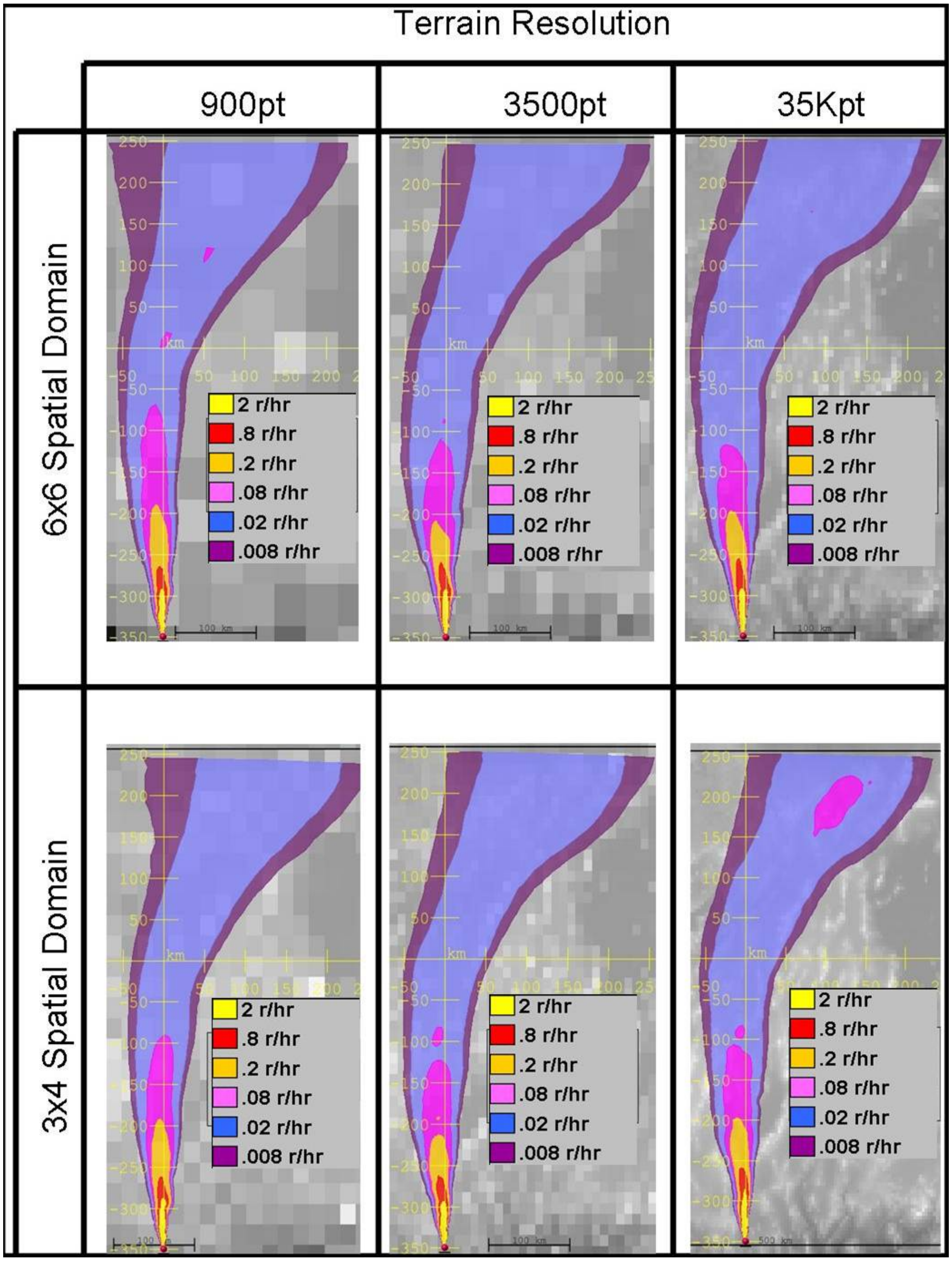


Figure 18 HPAC Simulations of George

represent terrain elevation with lighter shades of gray representing higher altitudes than darker shades. The simulation images show fallout extending to over 600 km from ground zero. This is in contrast to the DASA-EX data which only extends approximately

320 km. The extra length is kept in order to reveal visual differences between each simulation and is not used in the computation of numerical comparisons due to the lack of data in the original DASA-EX data. All HPAC simulation images reveal contour plots oriented to the north.   Upon closer inspection, it is evident that all simulation images have contour plot areas that lay slightly to the west of the vertical scale bar when considering only the first 300 km from ground zero. This is exaggerated in the no-terrain simulations. The DASA-EX plot shows that the contour plot area lay slightly to the east. To be more specific, if the centerline of the contours are considered, that line reside on the west side of the vertical scale bar for the simulation images while the DASA-EX image has the line residing on the east side. The protrusion of dose-rate contour area in the DASA-EX image is seen only in a subdued form on three of the six simulation images using terrain. The subdued protrusion appears on the east side of the plume near the -300 km marker. These are the 900/large, 900/small, and 3500/large simulations. It is clear that as the terrain resolution increases, the rendered elevation becomes a closer approximation to the actual terrain. It is interesting that as the simulation models terrain more accurately, the less apparent the protrusion feature becomes. Two possibilities exist for this behavior. First, the increased topographic gradient that comes with higher terrain resolution causes the mountain ridge to become more of a flow boundary than a deposition plateau. Second, the weather is changed by the atomic blast for a short time in a significant way as to slow the southerly winds to such a speed that causes settling on the mountain rather than flow over and around it.

Numerically, the simulations are compared to the observed data using the FORTRAN utility in Appendix E.   Though this research focuses on the NAD rather than the MOE, the MOE coordinates are directly related to the NAD value. For this reason, the MOE x- and y-coordinates are statistically studied with the Minitab Software Package [20] to reveal any underlying or supplementary information. Figure 19 is a graph of the NAD as differently parameterized simulations are compared to the DASA-EX observed data. For example, in the graph of NAD vs. Domain Size all of the simulations are

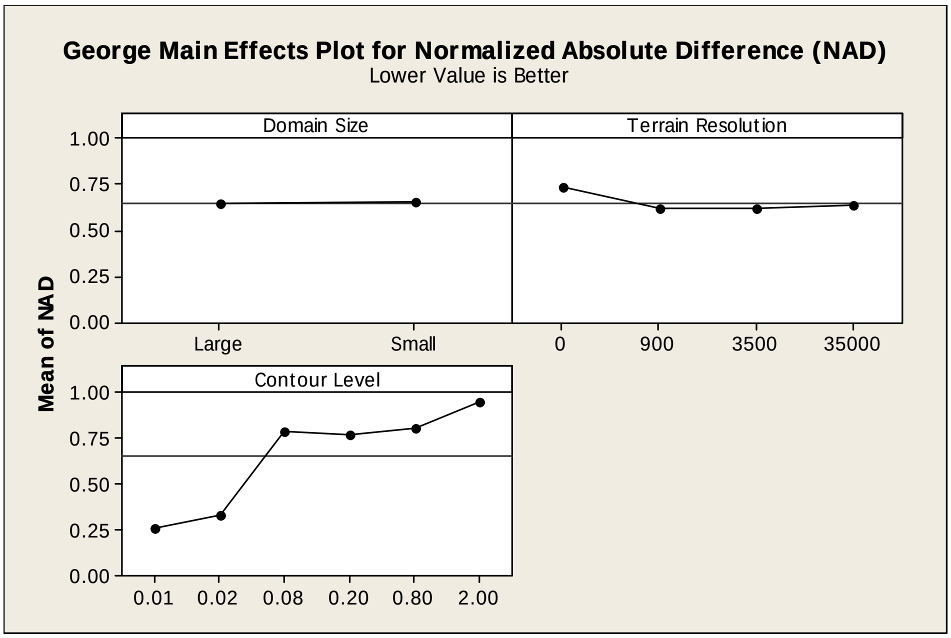


Figure 19 NAD vs Main Effects (George)

divided into two groups based on the domain size. Then the NAD is computed for every contour level regardless of terrain resolution or contour level. These values are then averaged and plotted. This allows domain size to be the only discriminator, or effect, in the computation of the NAD. In similar fashion, the NAD vs. Terrain Resolution graph divides the simulations into four categories based on terrain resolution.   NAD values are computed for every contour level, averaged, and then plotted. In the final graph of NAD vs. Contour Level, simulations are not divided. Instead, a NAD value is made for every contour level and then like-contour NAD values were averaged and plotted. For cases such as the NAD vs. Domain Size where there seems to be no difference in NAD values, the MOE coordinate values are plotted (See Appendix G for two-dimensional MOE Plots) to ensure that an x-coordinate value change in the positive direction does not counter a y-coordinate value change in the negative direction. The plots for MOE Coordinate vs. Main Effects are seen in Figure 20 and Figure 21. It is clear that for these main effects that the x- and y-coordinates of the MOE follow in exactly the same

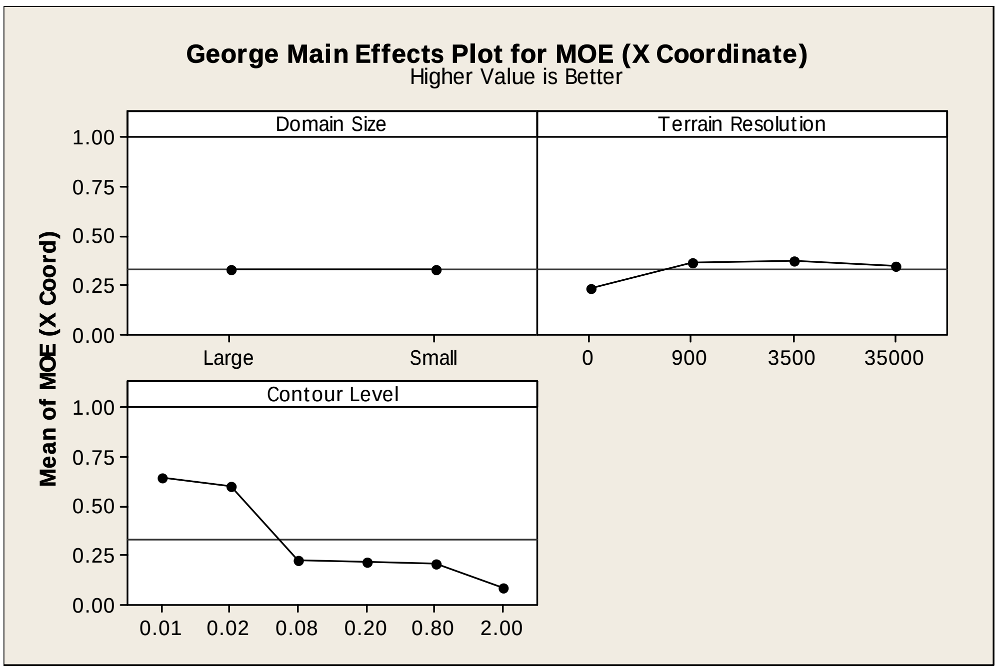


Figure 20 MOE x-coordinate vs. Main Effects (George)

fashion as the NAD. In terms of domain size, the NAD and MOE coordinates show indiscriminate differences between large and small domain sizes. The effect of terrain resolution seems to reveal a trend of increased accuracy when terrain is used but this trend seems to be reversed when terrain resolution increases beyond 900 points. Contour level seems to show a general increase in accuracy with a decrease in dose rate or, in other words, the lower the dose-rate contour level, the better the accuracy of the predicted model.

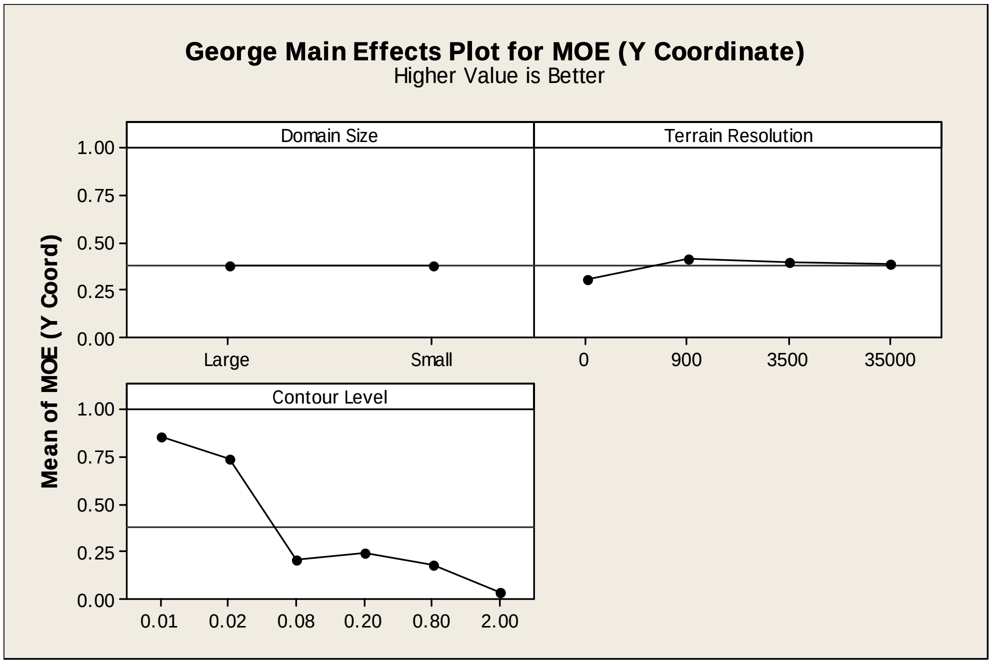


Figure 21 MOE y-coordinate vs. Main Effects (George)

In order to check the validity of these trends, an analysis of variance (ANOVA) is conducted on the data. Though the ANOVA is conducted with the Minitab software package, a cursory check on the assumptions behind this statistic is also conducted. In particular, the residuals are checked for normality as this is one of the premises upon which the ANOVA is based. A general linear model ANOVA considering three factors was computed using Tukey's method for statistical differences. Tukey's method is set up using the default 95% confidence interval. Though hardly conclusive, Figures 22, 23, and 24 illustrate a somewhat normal distribution of residual values. Other tests have similar results which are obtained using the Minitab software with the test data included in Appendix F.

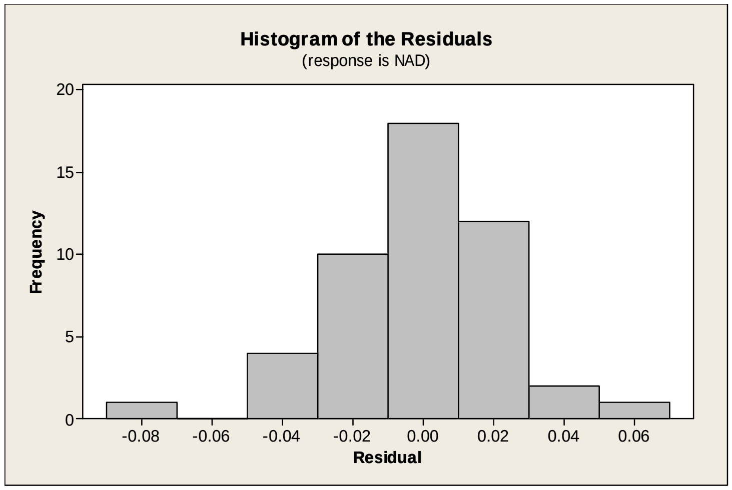


Figure 22 NAD Residual Plot (George)

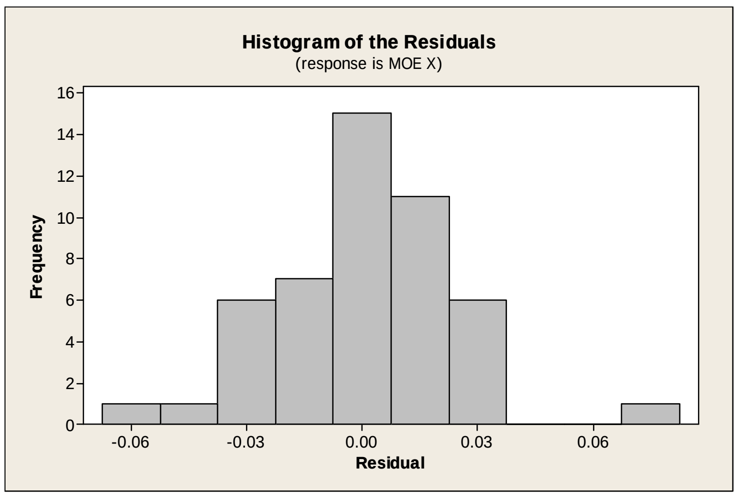


Figure 23 MOE x-coordinate Residual Plot (George)

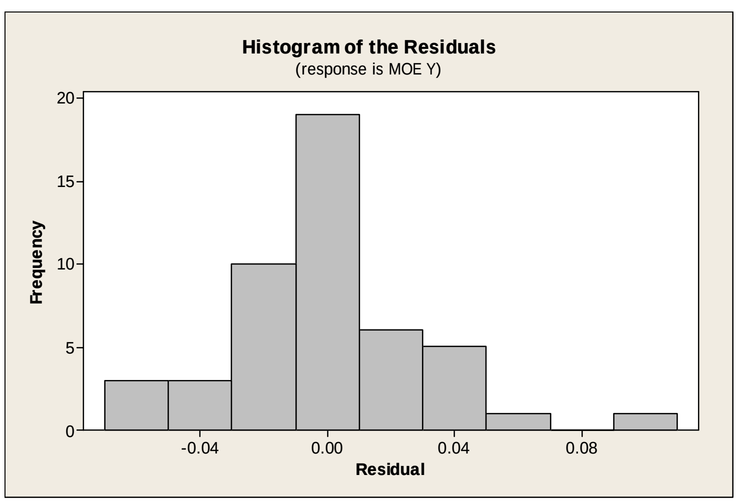


Figure 24 MOE y-coordinate Residual Plot (George)

The ANOVA confirms that there is no statistical difference in NAD or MOE coordinate values (0.65 p-value) when domain size is used as the discriminating factor. Also confirmed is the fact that accuracy makes a marked improvement between using no terrain and using terrain (0.00 p-value), however, the downward trend after 900 point terrain is not validated by the ANOVA. Though visually a downward trend can be seen, the fact is that the values at the 900-, 3500-, and 35K-point terrain resolutions are so close to one another that the differences are not significant enough to rise above the statistical noise. This is true for both the NAD and both MOE coordinates. The ANOVA statistic for the contour-level factor indicates that the 2 r/hr contour line is indeed of lesser accuracy than all other contour levels. The test statistic also indicates that the .008 r/hr contour line is of higher accuracy than the other contour intervals. Accuracy of the .02 r/hr contour level is statistically the second most accurate. However, the .08, .2, and .8 r/hr contour lines are statistically of the same accuracy and therefore cannot be rank ordered.

Operation Teapot – Ess

The DASA-EX document describes the same six off site dose-rate contours as the George shot. The area in which contour data was taken extends approximately 200 miles downwind of ground zero. Figure 25 illustrates a northerly-oriented digitized representation of the DASA-EX contour plot. This contour plot is characterized by a generally south-easterly flow out to approximately 100 km followed by an easterly smearing occurring out to 210 km due east.

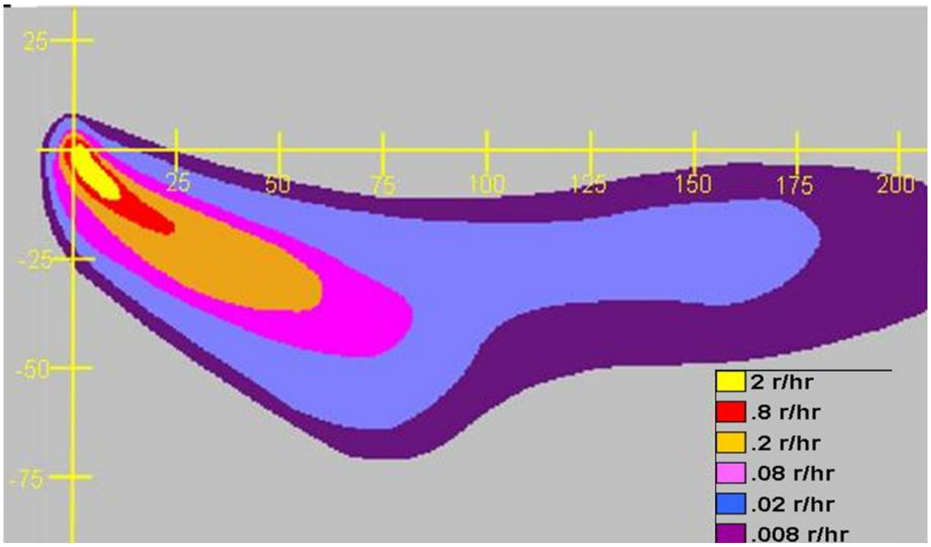


Figure 25 Ess Digitized DASA-EX Contour Plot

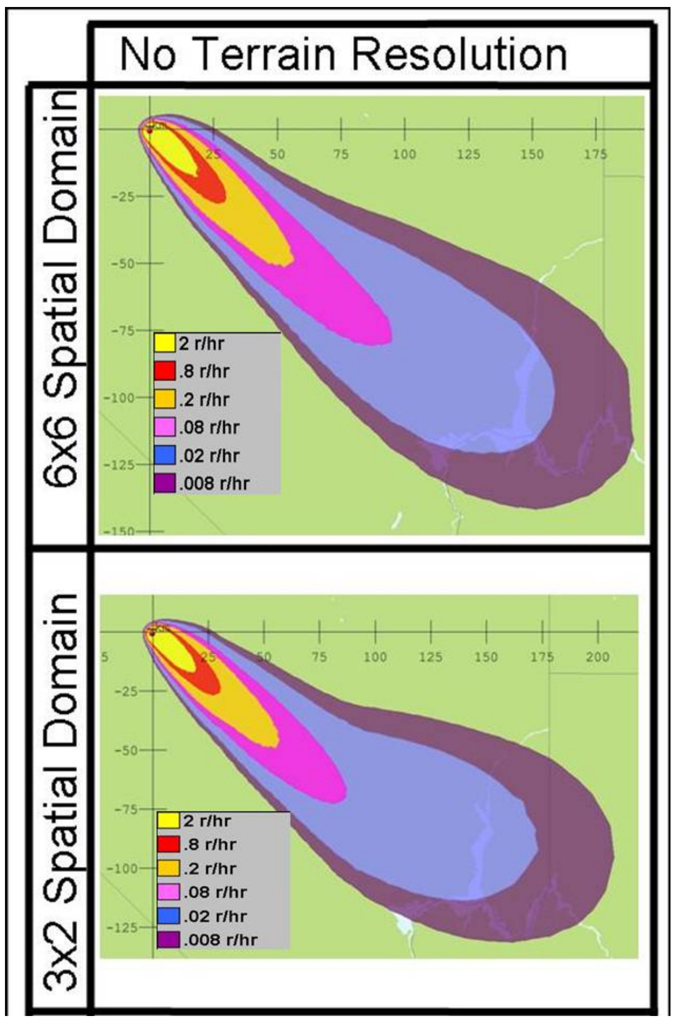


Figure 26 Ess Simulations Using No Terrain

HPAC simulations of the Ess shot using no terrain (Figure 26) both demonstrate the south-easterly flow but fail to show any significant smearing in the easterly direction. Both of these simulation images, compared to each other, show approximately the same south-easterly contours with only slight variation in shape and distances traveled. The Ess simulations using terrain show significant changes as the terrain resolution increases,

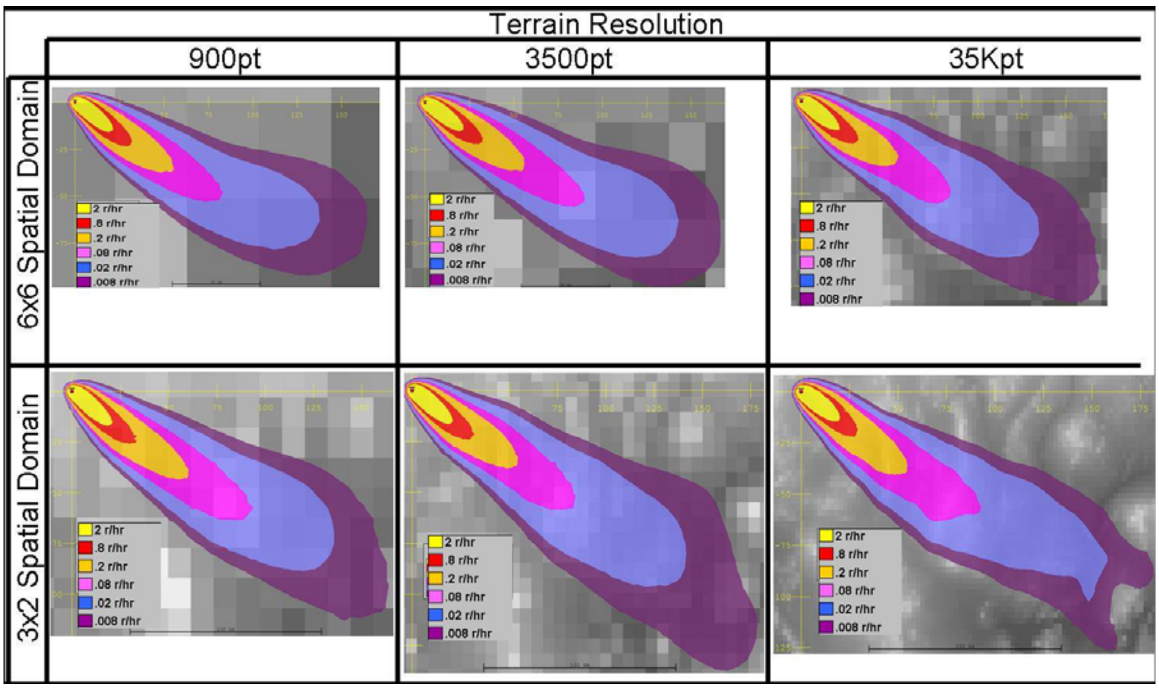


Figure 27 Ess Simulations Using Terrain

specifically in the small/35K simulation. In this image, the extreme south-east edge of the contours seems to flow around a mountain and through adjacent valleys. This is a positive indicator that HPAC is attempting to physically model airflow changes due to topographic relief. Interestingly, these flow dynamics only seem to be revealed at affected areas furthest from ground zero. One possible explanation for this phenomenon is that the flow dynamics did indeed occur at all locations of the affected area and that the effect is only apparent when the settling fallout is of the smallest particle sizes.

The numerical agreement between the simulations and the observed data is seen in Figures 28, 29, and 30. Initial observations of the NAD data points seem to show no difference due to domain size but perhaps a difference with reference to terrain resolution. The contour-level factor seems to reveal a pattern that lends accuracy toward the use of the middle two contour levels, namely, .08 and .2 r/hr. The MOE coordinates

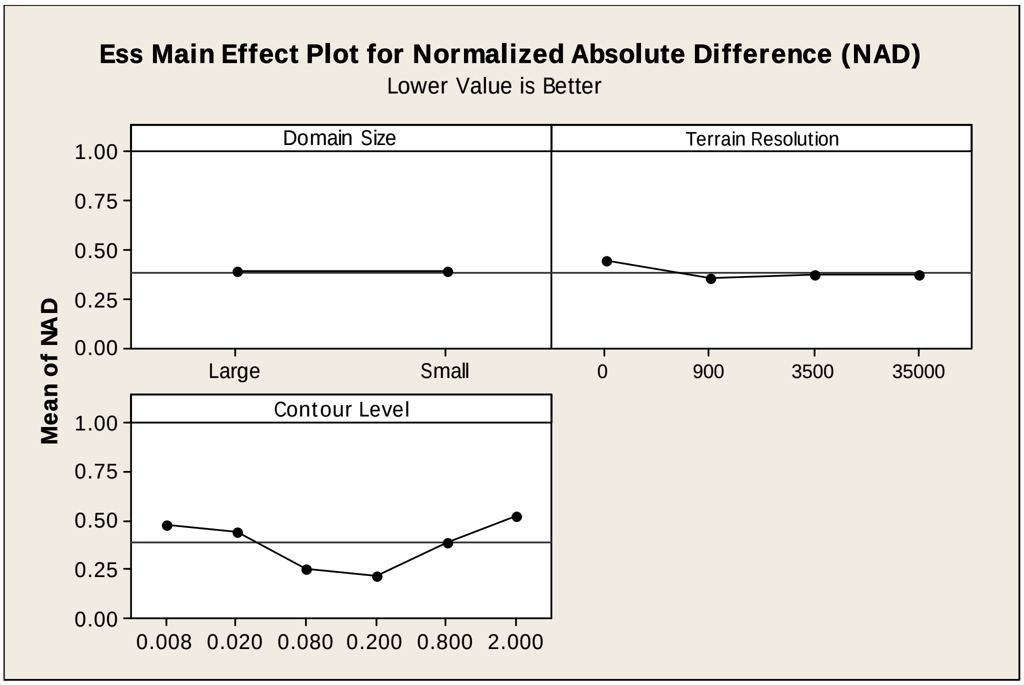


Figure 28 NAD vs. Main Effects (Ess)

show the same pattern as the NAD values for both the domain size and terrain resolution factors with the MOE y-coordinate showing a much more significant improvement from 0- to 900-point terrain. The contour-level factor for the MOE x-coordinate shows a general trend of improvement with increasing contour levels. In contrast, the MOE y-coordinate displays the same general accuracy trend as the NAD values.

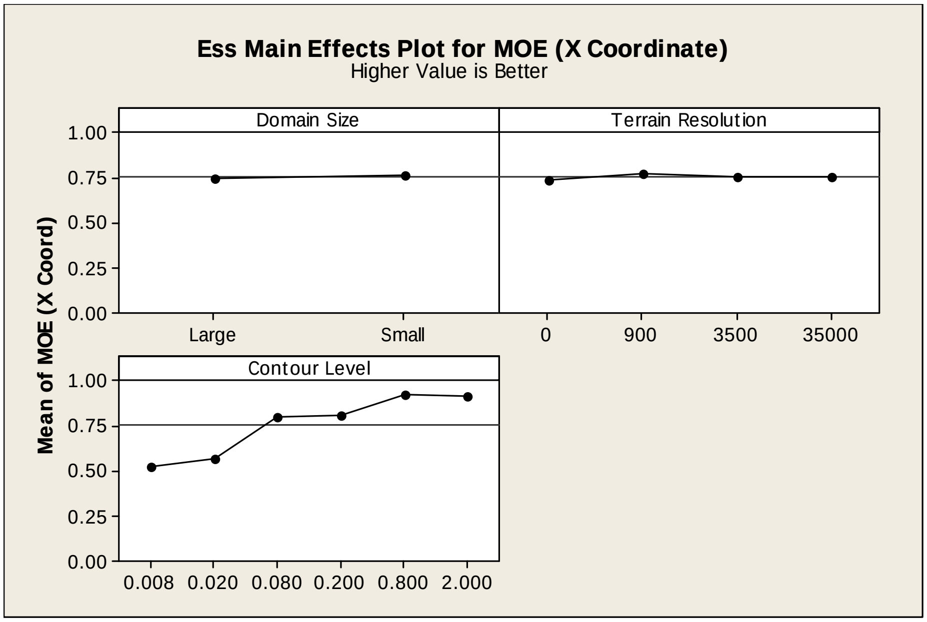


Figure 29 MOE x-coordinate vs. Main Effects (Ess)

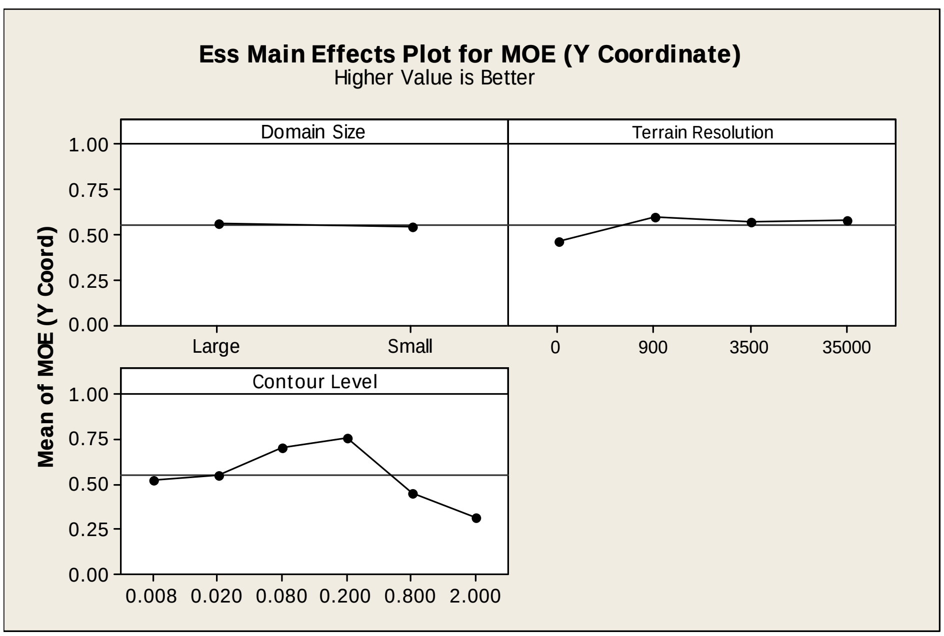


Figure 30 MOE y-coordinate vs. Main Effects (Ess)

The ANOVA confirms that the NAD (0.88 p-value) and MOE y-coordinate show no statistical difference in values in terms of domain size. However, the MOE x-coordinate does show that a small domain size has a distinctly higher accuracy than the large domain size. The terrain resolution factor shows no significant accuracy difference for the MOE x-coordinate but the analysis does show that both the NAD (.01 p-value) and MOE y-coordinate have a lower accuracy when simulations are carried out using 0-point terrain. There is no difference, however, between the 900-, 3500-, and 35K-terrain simulation values. Finally, the analysis shows that the NAD and MOE y-coordinate values, when contour level is taken as the main effect, separate the contours into two groups. The middle two contours, .08 and .2 r/hr, are more accurate than the other four contour levels. It is impossible, however, to indicate which contour is more or less accurate than another when performing intra-group comparisons. In contrast, the MOE x-coordinate shows a general increase in accuracy for higher contour levels with the highest two contour levels being more accurate than the middle two, and the middle two being more accurate than the lowest two contour levels. Again, within groups, each contour level has no distinctly higher accuracy than the other.

Operation Teapot – Zucchini

The DASA-EX document describes the same six off site dose-rate contours as the George and Ess shots. The digitized contour plot is seen in Figure 31. This contour plot is characterized by a generally south-easterly flow for approximately 128 km and then turning to the north-east for an additional 190 km. Upon examination of the plot, there are two areas of particular interest. The first is the pocket of unaffected area located 200 km east of ground zero. This geographic area is characterized by a mountain ridge positioned along the western edge of the pocket (See Figure 32). I believe that the mountain ridge acts as a wind ramp causing updrafts. This updraft results in a fallout shadow where particles essentially skip over the non-contaminated area. The second area of interest is the small oval area enclosed by a .08 r/hr contour level.

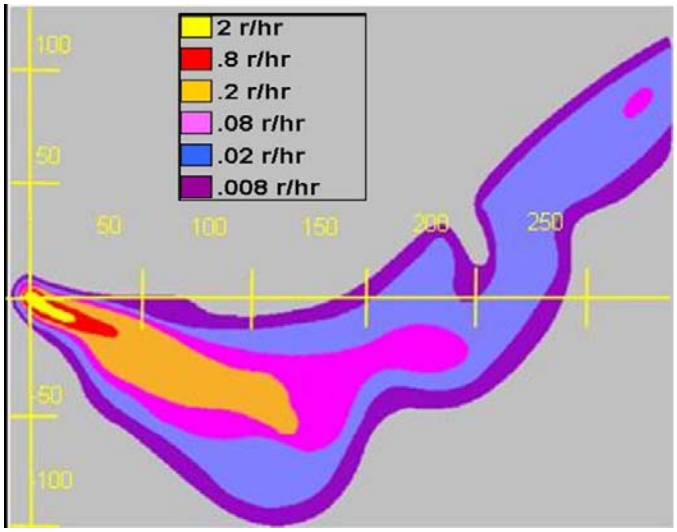


Figure 31 Zucchini Digitized DASA-EX Contour Plot

This oval is located near the eastern edge of the observed area and is surrounded by a .02 r/hr contour area. By overlaying the DASA-EX image onto Google Earth, the area in

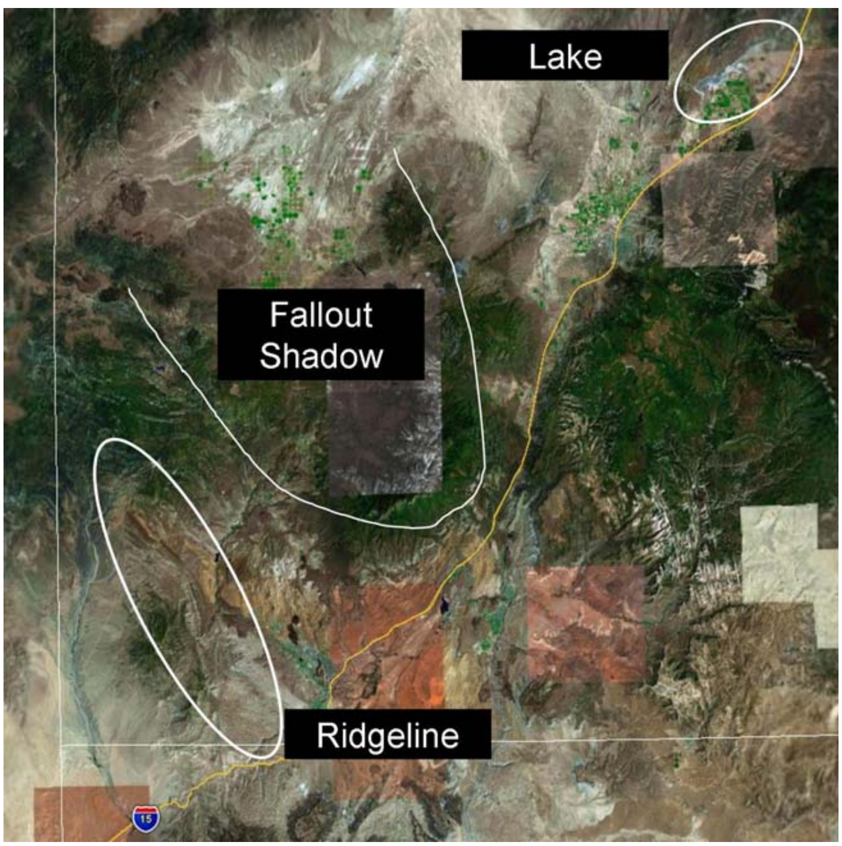


Figure 32 Terrain (Zucchini)

question is found to be a lake. The higher dose rate can be explained by the fact that as the blanket of radioactive particles settled near the lake, they were more likely to remain in place as they lodged into the moist, and relatively sticky, soil. Another possibility is that some small particles entered the relatively humid air and absorbed water vapor causing increased mass and volume resulting in a quicker descent to the ground. It is unknown if the surveyors took water surface measurements from boats. An assumption is made that the surveyors did not take measurements three feet above the lakes surface. If they did, measurements taken would have been much lower than surrounding land areas because particles landing on the water's surface would be consumed by the volume of the lake and therefore the detectors would have been somewhat shielded by the physical properties of the colloidal system.

Figure 33 and Figure 34 display the simulation results using varying degrees of terrain resolution. All simulations produce extremely similar plots out to a distance of 250 km east of ground zero. The most obvious difference between the simulations and the DASA-EX image is the lack of a south-east contour flow. The only conclusion that can be drawn from this observation is that the reanalysis weather data does not capture the entire wind patterns for this detonation because the spatial and/or temporal resolution of the reanalysis weather is too coarse for accurate modeling.

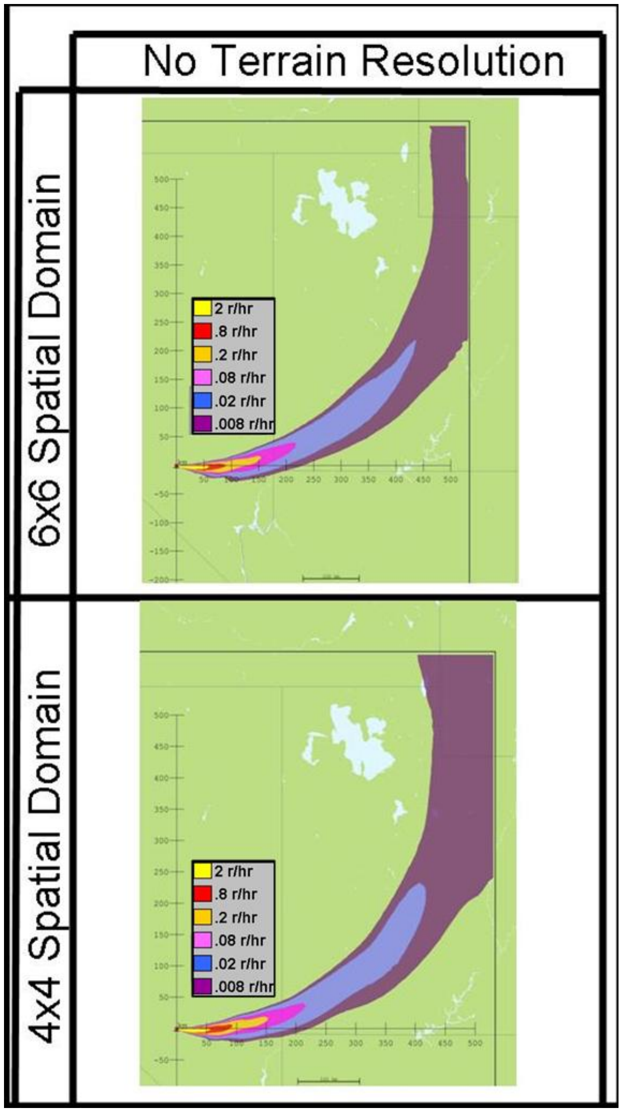


Figure 33 Zucchini Simulations Using No Terrain

The major differences between simulation images themselves only manifest near the north-west corner of the spatial domain. Because the DASA-EX data is limited to 250 km east of ground zero these differences do not influence the numerical comparison. It is interesting to note that as terrain resolution increases, artifacts of fallout deposition appear in the form of terrain-influenced contours. For example, the small/35K simulation shows an exaggerated rippling effect of the .02 r/hr contour seemingly caused by a ridgeline. This same rippling effect can be seen in the .008 r/hr contour at the base, or southern valley, of this same ridgeline. The small/900 simulation contains a break in the contour plot. This could be HPAC's weather model simulating the same ramp-like phenomena as seen in the digitized Ess DASA-EX plot. For the small/3500 and the small/35K simulations, the skipped area shows a broadening of the contours. This fits well with the ramp hypothesis as the low 900 point resolution causes a steeper gradient leading to the summit of the mountain. This steep gradient lends itself to speedy vertical

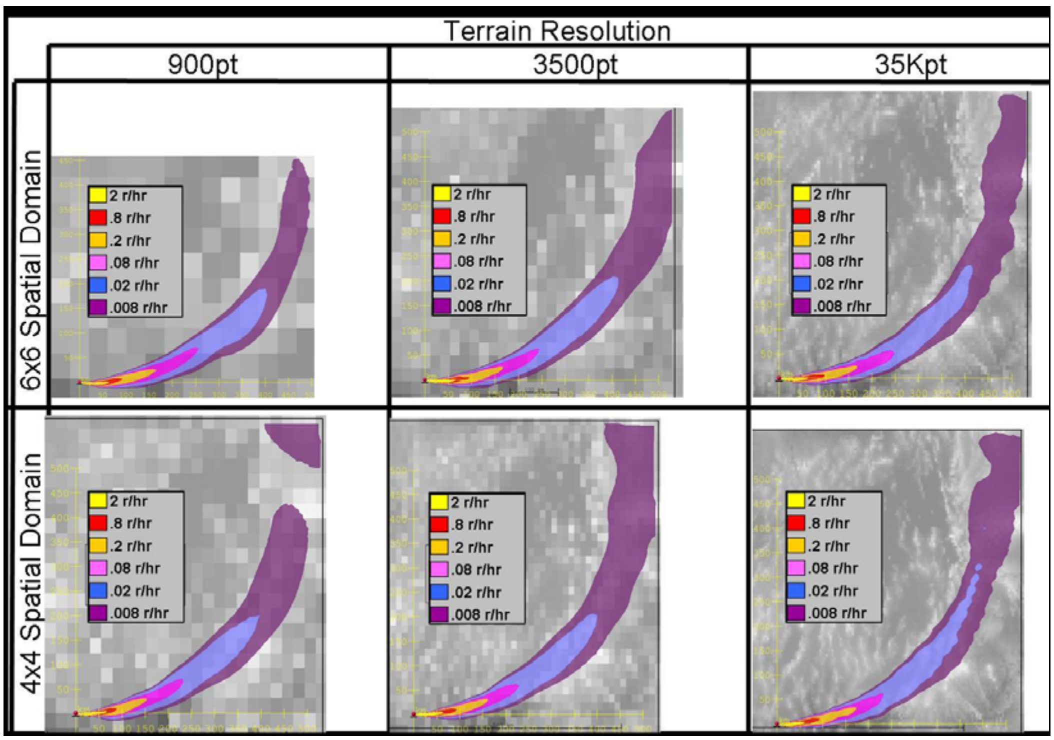


Figure 34 Zucchini Simulations Using Terrain

winds that could produce a fallout shadow.

Figures 35, 36, and 37 all show similar results. All three figures show that accuracy is favored by a large spatial domain and the use of no terrain. There is also a general trend of increased accuracy with lower contour levels with the exception of the MOE x-coordinate. The MOE x-coordinate shows a subdued cupping pattern with accuracy at its lowest for the .08 r/hr contour level. After that point, the trend seems to show rising accuracy with increasing contour levels.

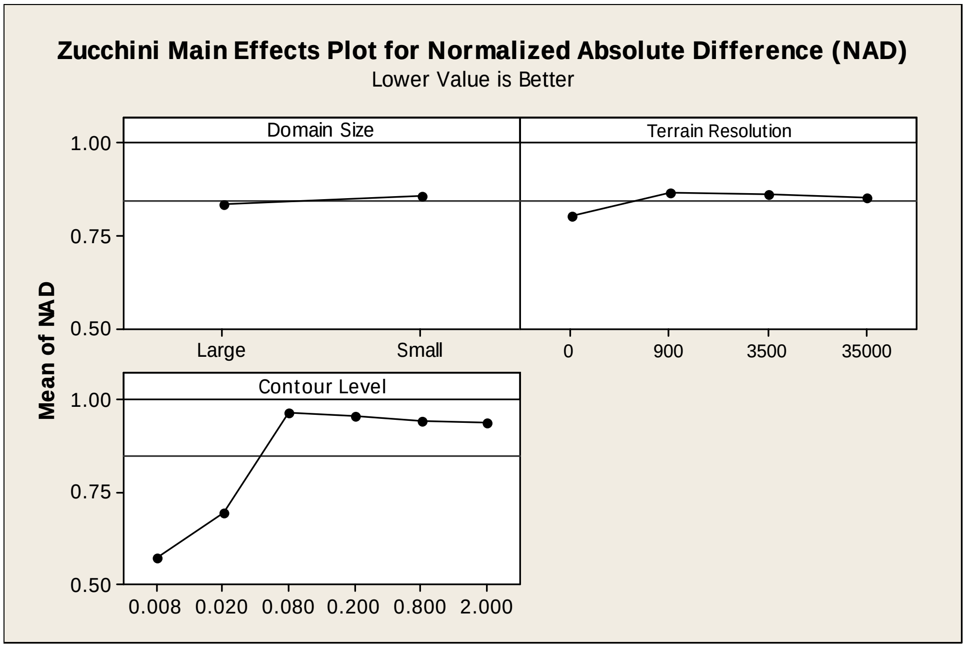


Figure 35 NAD vs. Main Effects (Zucchini)

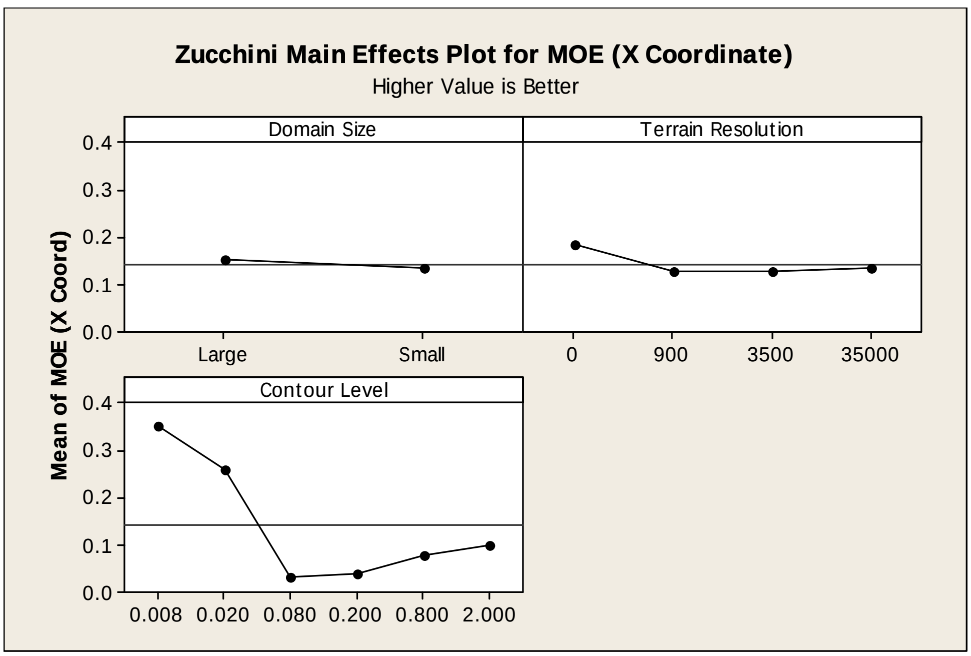


Figure 36 MOE x-coordinate vs. Main Effects (Zucchini)

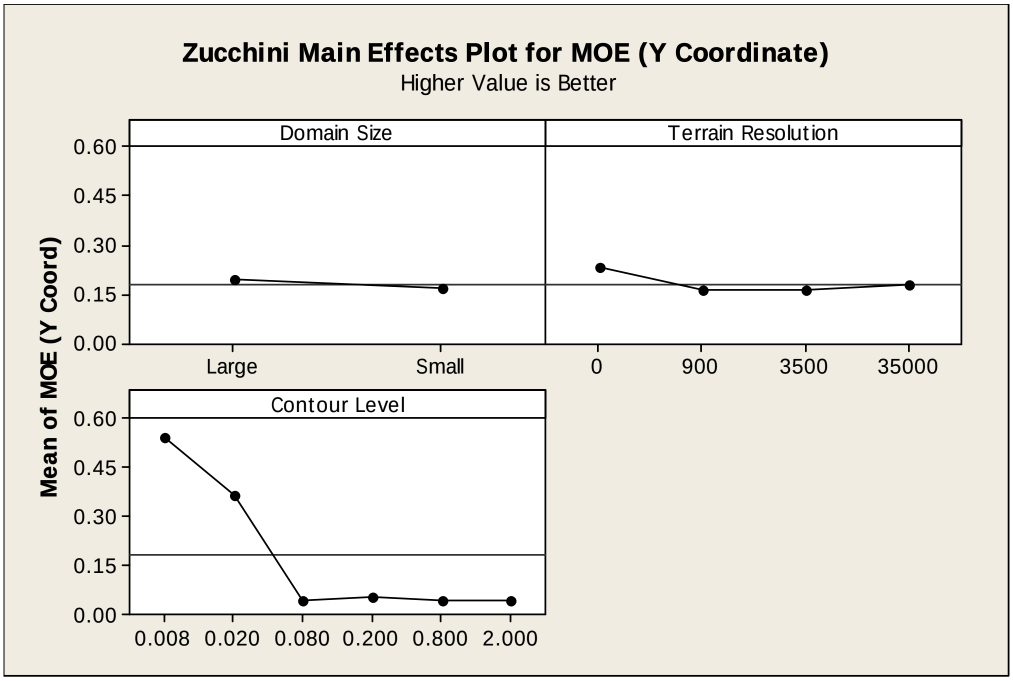


Figure 37 MOE y-coordinate vs. Main Effects (Zucchini)

The ANOVA reveals that using a large spatial domain leads to a more accurate prediction (0.01 p-value). Also, just as visually observed, using no terrain results in a more accurate prediction than not using terrain (0.00 p-value). The terrain resolution to use, however, for the least accurate prediction cannot be determined. In both the NAD and MOE y-coordinate graphs, the contour levels are ranked in order of accuracy with the .008 r/hr contour level being the most accurate followed by the .02 r/hr level. The other four levels cannot be distinguished in terms of accuracy from one another but are determined to be of less accuracy than the lowest two contour levels. The MOE x-coordinate shows the same trend for the .008 and .02 r/hr levels but the last four contours can be divided. The ANOVA, in this case, indicates that the .08 and .2 r/hr contours are the least accurate. The .8 and 2 r/hr contours are of higher accuracy than the .08 and .2 r/hr contours but are less accurate than the .02 r/hr contour. Within groups, the contour levels are indistinguishable in terms of accuracy.

Operation Plumbbob – Priscilla

The Pricilla test fallout pattern is characterized by only four contour lines whose easterly downwind pattern can be viewed in Figure 38. This contour pattern is

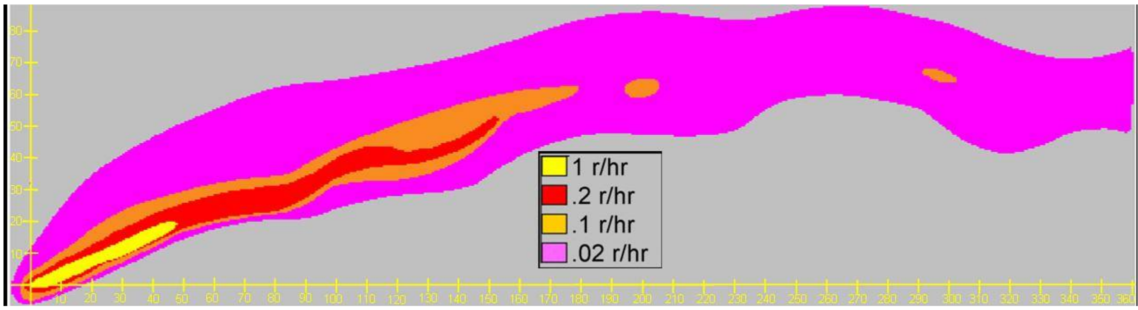


Figure 38 Priscilla Digitized DASA-EX Contour Plot

unremarkable except for two oval areas of .1 r/hr dose rates located 200 km and 300 km east of ground zero. These western-most oval can be explained by a topographical ramp causing local updrafts. This ramp consists of a relatively wide valley running perpendicular to the wind direction approximately 50 km upwind, near the tail of the main .1 r/hr contour area. This valley can cause horizontal vorticity resulting in an updraft along the canyon walls. The second oval is located in a large valley just downwind of a large, high plateau-like feature. Unlike the first oval, this valley is marked by a higher activity level which supports the idea of local horizontal vorticity causing higher deposition rates and thus higher dose rates.

Figure 39 shows almost identical results for simulation runs using no terrain. There is, however, a small area of deposition beyond the main plume to the east of the

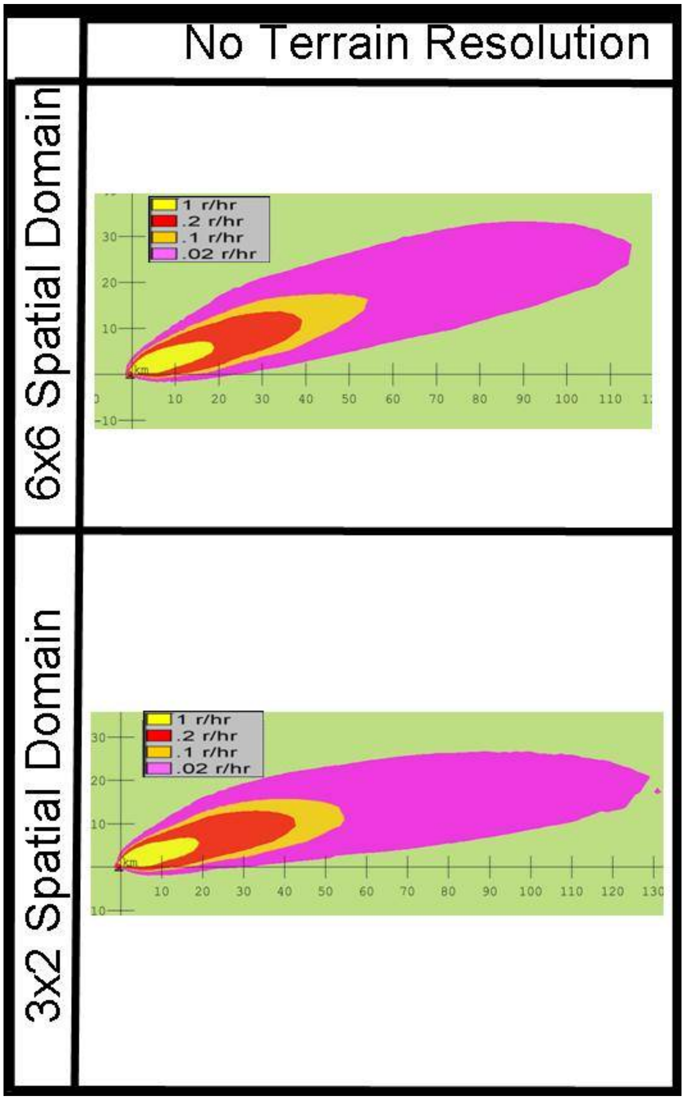


Figure 39 Priscilla Simulations Using No Terrain

simulation using a small spatial domain. In both cases, the direction is in keeping with the DASA-EX contour plots but fall extremely short in terms of overall distance of the contours.

Figure 40, just as in Figure 39, shows little variability in the simulation runs even though varying terrain resolutions were incorporated. Just as before, the images depict

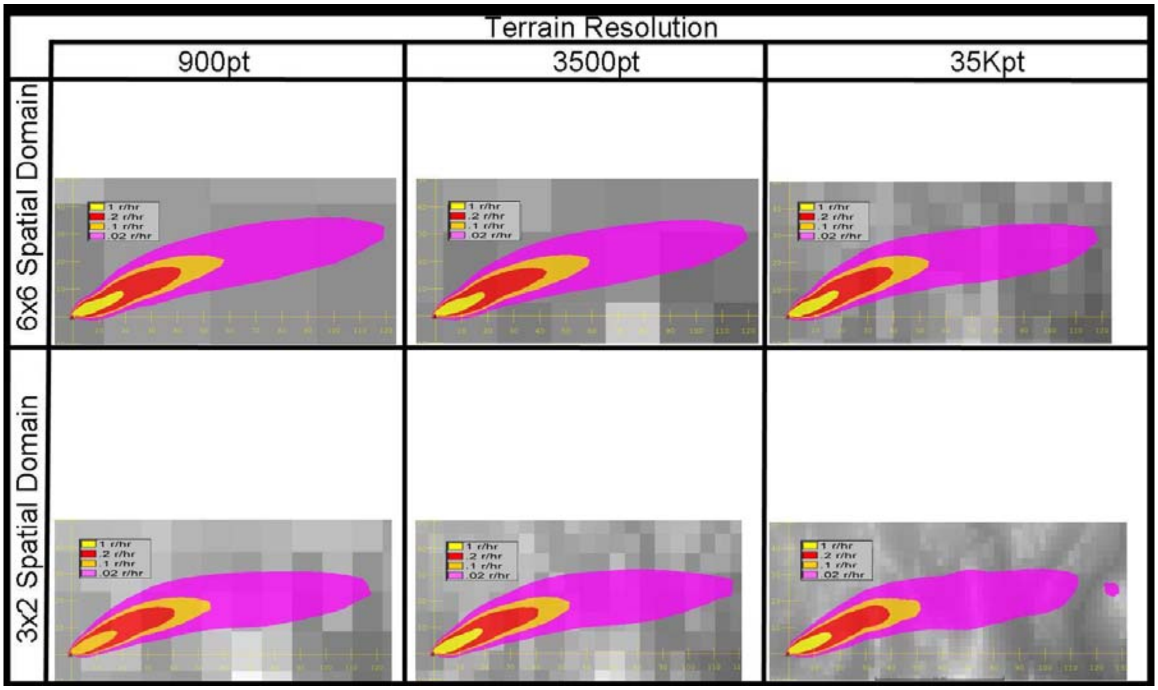


Figure 40 Priscilla Simulations Using Terrain

a similar plume direction as the DASA-EX image while failing to achieve even half of the recorded distance as the original test data. The small area of deposition to the east of the small/35K simulation image is similar to the small simulation run in Figure 39. This is interesting as the mountain top in the small/35K simulation can be attributed to the isolated activity while the small/no-terrain simulation can attribute the pocket to no such feature.

Figures 41, 42, and 43 all show a possible accuracy advantage when the small terrain is used in the HPAC model. All figures also show a possible decrease in

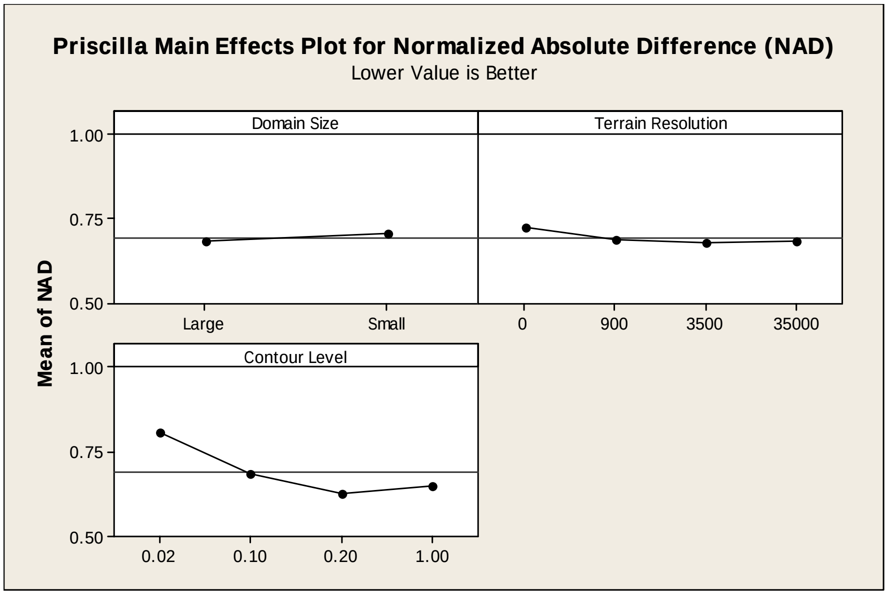


Figure 41 NAD vs. Main Effects (Priscilla)

accuracy when using terrain. Finally, both the NAD and MOE x-coordinate display a general accuracy improvement given higher contour levels. In contrast, the MOE y-coordinate shows that mid-level contours are favored.

The ANOVA reveals that domain size plays no significant role in model accuracy (.11 p-value for NAD). The analysis concludes that for the NAD (.06 p-value) and MOE x-coordinate, terrain resolution plays no significant role. This contrasts with the MOE y-coordinate where terrain does improve accuracy, though which terrain resolution above zero renders the most accurate model cannot be identified. The NAD and MOE x-coordinate are confirmed in the fact that higher resolution results in a more accurate HPAC simulation. Just as before, the .2 and 1 r/hr contours cannot be distinguished from one another in terms of accuracy. The MOE y-coordinate does indeed show that the .1 and .2 r/hr contour levels are more accurate, as a group, than the .02 and 1 r/hr contour levels which are also individually indistinguishable in terms of accuracy advantage.

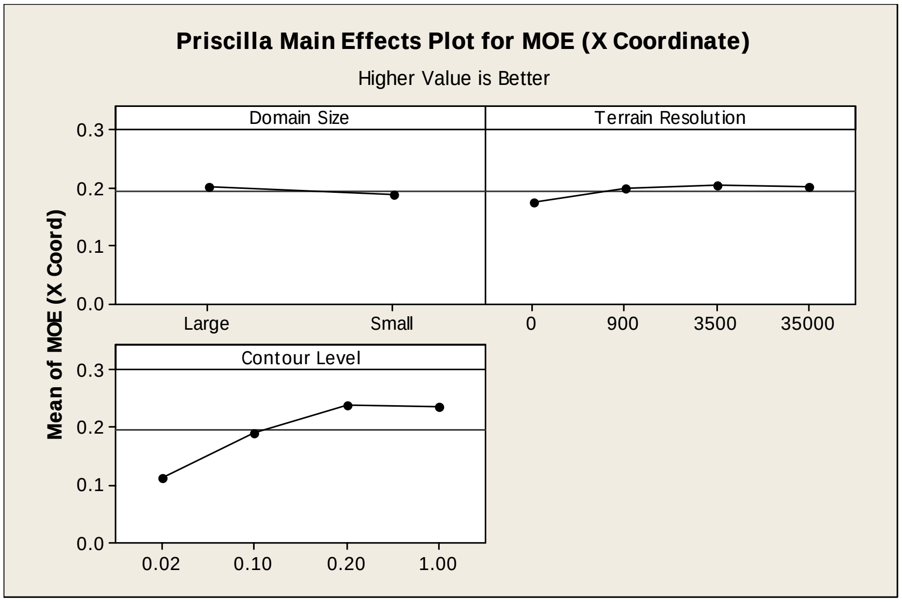


Figure 42 MOE x-coordinate vs. Main Effects (Priscilla)

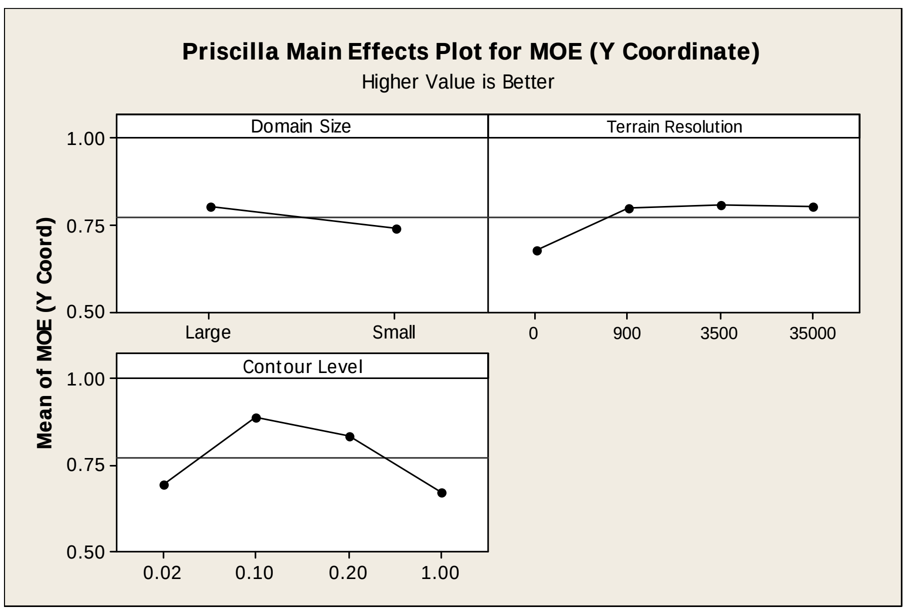


Figure 43 MOE y-coordinate vs. Main Effects (Priscilla)

Operation Plumbbob – Smoky

The Smoky contour plot from the DASA-EX document contains seven contour levels, the most of any test in this research. The fallout survey reveals a plume (See Figure 44) that initially flows in a south by south-easterly direction and after about 130 km shifts almost exactly north-east. During the initial south-east flow, all contour levels are somewhat bunched to the north with respect to the .02 r/hr contour level. As the path shifts to the north-east, the .2 and .1 r/hr contour levels proceed to be shifted to the southern edge of the .02 r/hr contour level. The feature of interest in this plot is the forked tail of the .1 r/hr contour level. The northern fork is positioned over a mountain ridge which could explain its higher dose rate in terms of ability to catch lighter particles aloft. The southern fork, however, does not seem to have any key topographic characteristics that make it more likely to receive a higher amount of activity than the surrounding area.

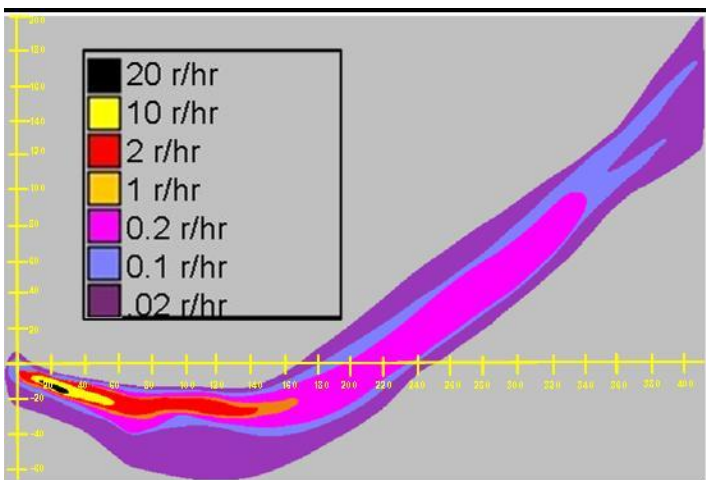


Figure 44 Smoky Digitized DASA-EX Contour Plot

Figure 45 and Figure 46 show that almost every simulation has the exact same contour direction. However, the lack of the initial south-easterly flow indicates that either key weather information was missed due to the coarse spatial and temporal domains of the reanalysis weather or that there is a problem with HPAC's cloud rise model. A problem in the cloud rise model's calculation would put the stabilized cloud at the wrong altitude and therefore subject to a possibly vastly different set of weather conditions. According to Glasstone and Dolan's stabilized cloud chart on page 431, Smoky's stabilized cloud would have had a bottom at about 15,000 ft and a cloud top at almost 30,000 ft. The weather file created from reanalysis weather indicates that these heights equate to between 100 and 10 mb respectively. Using HPAC's weather profile viewing utility, it is clear that none of these heights have the correct wind direction. In fact, at altitudes higher than 15,000 ft, reanalysis weather contains winds that are blowing against the plume direction. However, there are required wind directions at nearby reanalysis-defined weather locations at the 400 mb, or 7000 ft, level. These facts lead me to conclude that the answer to the missing south-east dip probably includes both coarseness of weather data and cloud rise calculation error.

Another difference between the simulation images is that models using no terrain extended approximately 270 km from ground zero while the simulations using terrain ranged from 320 to 370 km[8].

[8] The small/35K simulation initially extended beyond the set spatial domain resulting in a truncated plume. In order to make numerical comparisons, the spatial domain was extended by 2.5 degrees of longitude. Though this results in two different small spatial domains, the statistical analysis only considers 'large' and 'small' and thus the change does not disturb the final result.

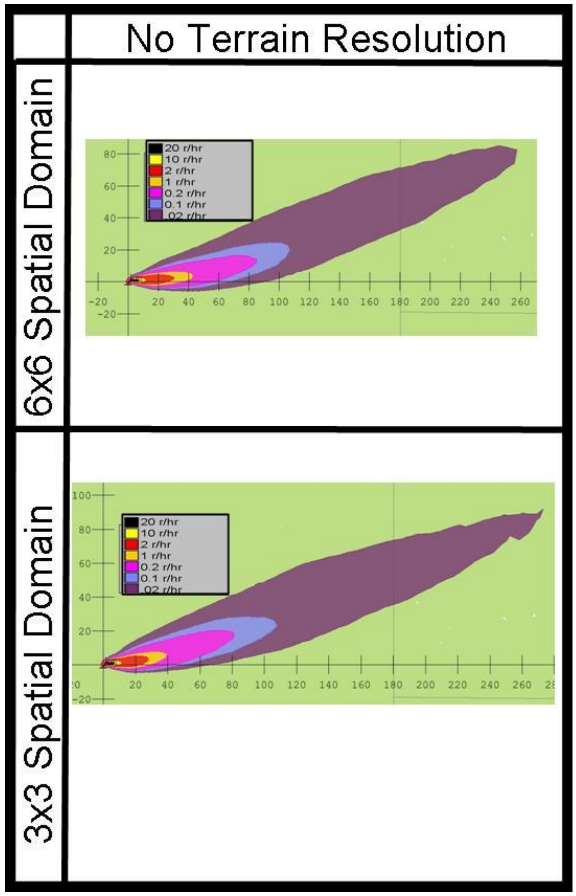


Figure 45 Smoky Simulations Using No Terrain

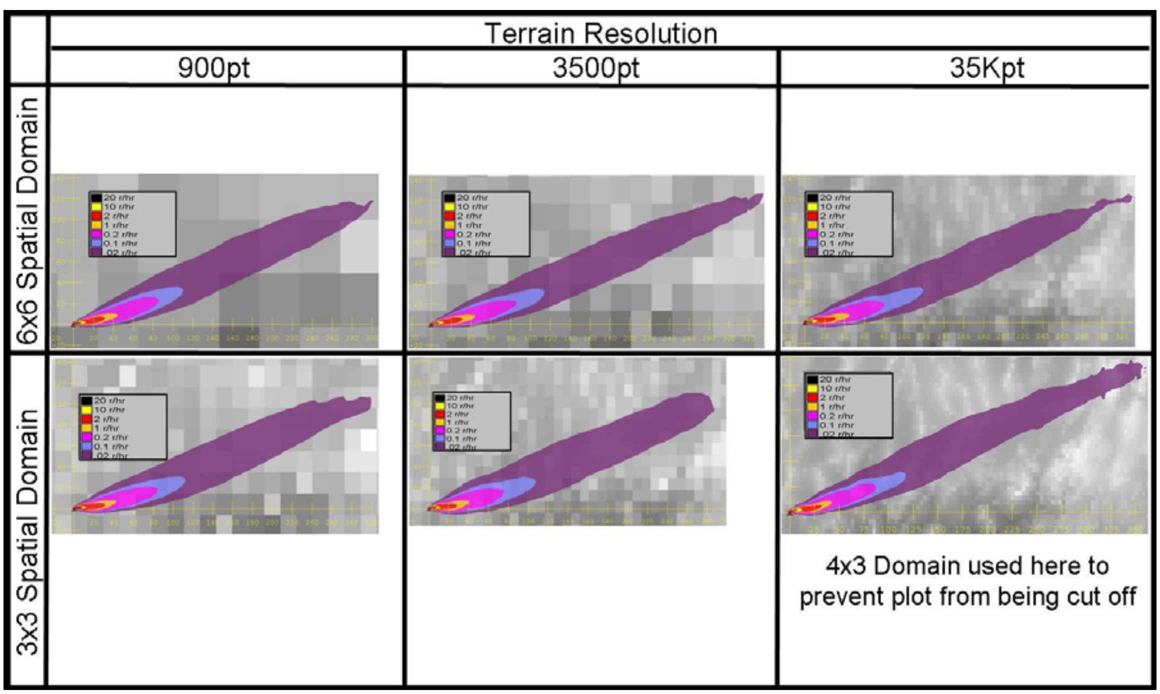


Figure 46 Smoky Simulations Using Terrain

Though an analysis is completed for the Smoky detonation, it is of purely academic use. The scale of the graphs contained in Figures 47, 48, and 49 are of such limited range that statistical significance is of little practical value. Even so, there is academic value in the analysis as it aids in the identification of trends.

All figures visually point to a domain and terrain resolution as being a non-discriminatory factor in accuracy. When viewing the contour level graphs, it seems that if the .10 r/hr contour line were taken as an anomaly, all graphs would point to a higher

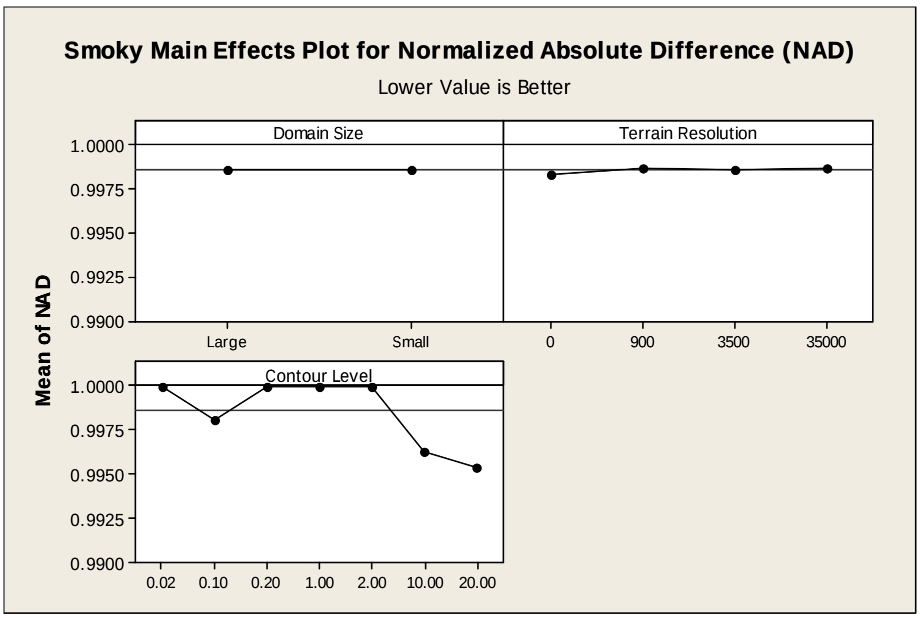


Figure 47 NAD vs. Main Effects (Smoky)

accuracy being associated with higher contour levels. Given the fact that the simulation plots miss the initial south-east directionality, it is understandable that the higher contour levels would be of greater accuracy. This is due to higher contour levels having a shorter range. The conclusion being that given a simulation using the wrong weather, there is less time for the higher contour plumes to be misguided before deposition.

The ANOVA confirmed the visual observation that domain size does not affect accuracy (.68 p-value for NAD). It further confirmed that terrain resolution plays no significant part in terms of the NAD (.07 p-value) and MOE x-coordinate. However, the MOE y-coordinate shows that the 35K point terrain resolution is of significantly lower accuracy. The 0-, 900-, and 3500-point resolutions could not be distinguished from each other in terms of accuracy. The contour

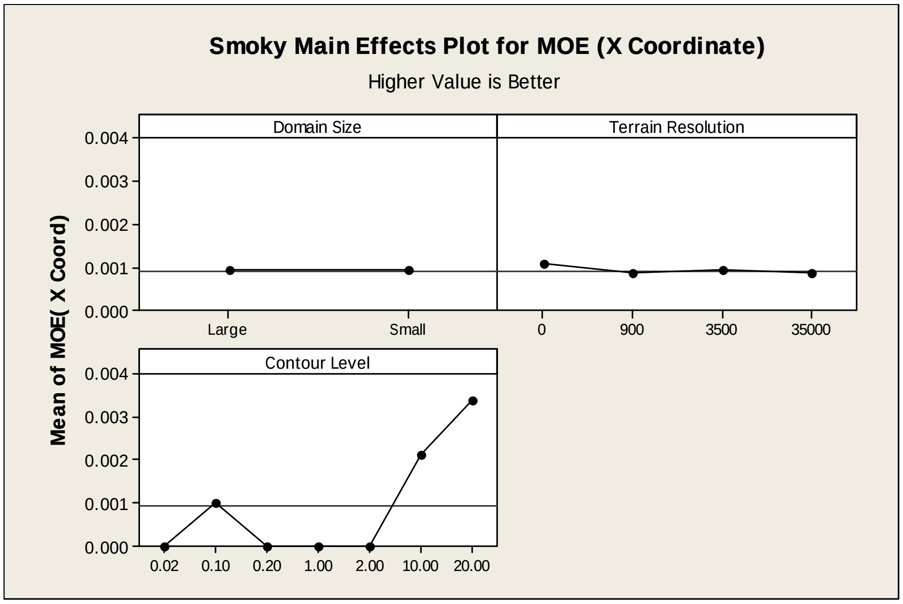


Figure 48 MOE x-coordinate vs. Main Effects (Smoky)

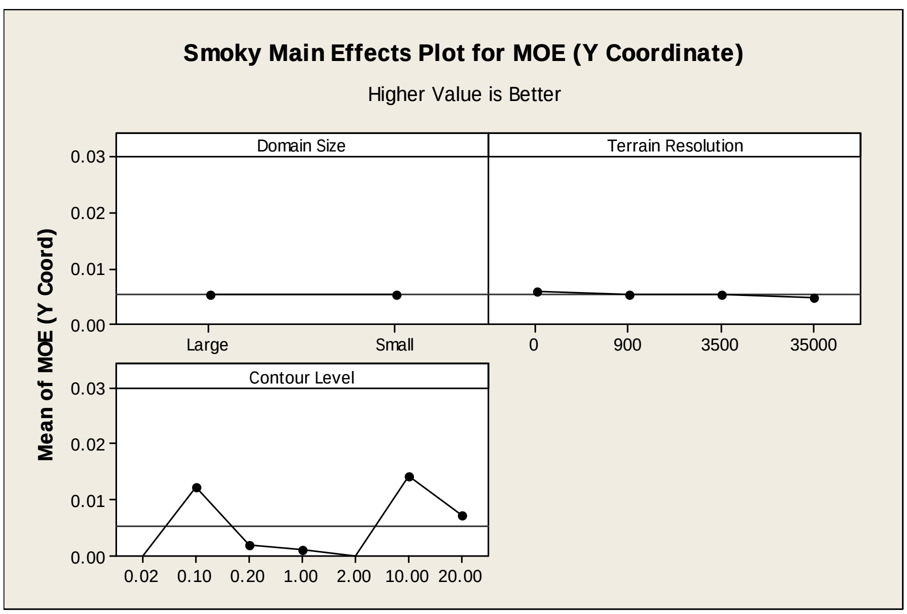


Figure 49 MOE y-coordinate vs. Main Effects (Smoky)

levels do show some differences in accuracy though a trend is not discernable. The NAD and MOE x-coordinate contour-level graphs have the same statistical groupings. The lowest accuracy group, which cannot be distinguished from each other, contains the .02, .2, 1, and 2 r/hr contour levels. The next most accurate contour level is the .1 r/hr dose rate followed by the 10 r/hr level. The 20 r/hr contour level is the most accurate. The MOE y-coordinate graph is only distinguishable as the 10 r/hr is more accurate than the 20 r/hr contour level. This confirms the visual observation that, except for the .1 r/hr dose rate, higher contour levels generally coorelates to accuracy.

Operation Sunbeam – Johnie Boy

The Johnie Boy fallout pattern is characterized by six contour levels. The fallout pattern depicted in Figure 50 is northerly directed and indicates a slight initial westerly flow followed by a slight shifting to the east. The DASA-EX data was truncated at approximately 75 miles north of ground zero. Ground zero for Johnie Boy is located about midway up the side of a mountain. To the north of ground zero the peak of the mountain becomes a north-running ridgeline. This ridgeline extends beyond the plume boundaries. The plume seems to run along, but not on top of, the ridgeline. The .5, 1, and 10 r/hr contour lines flow to the west into a valley. The .1 and .05 r/hr contours then

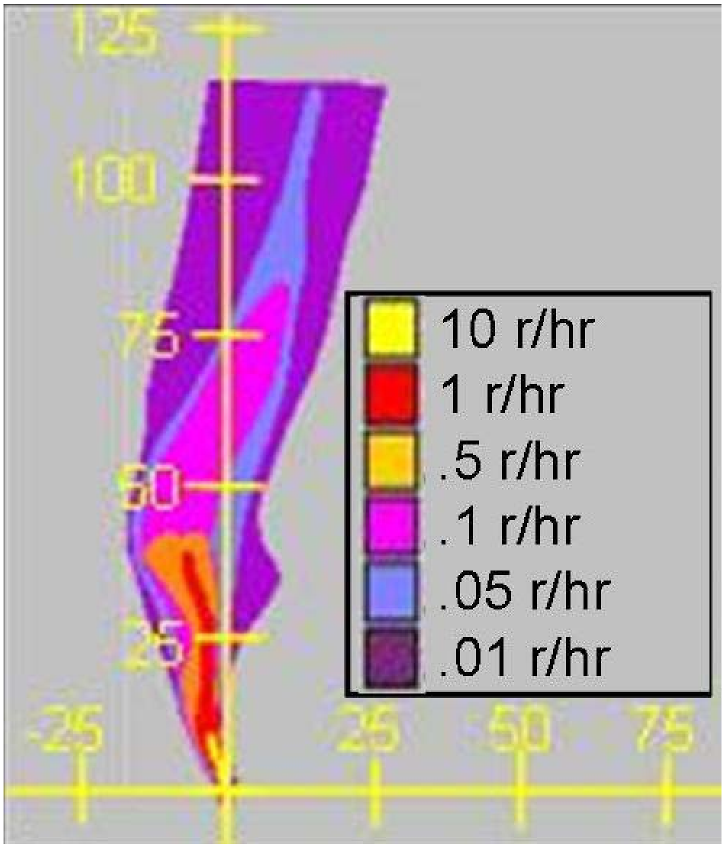


Figure 50 Johnie Boy Digitized DASA-EX Contour Plot

begin to flow somewhat to the east flowing over the ridge, across a valley and then finally settling on another ridgeline. As the higher contours are typically made from larger particles, we see that these particles settled quickly and were carried by the wind around the mountain. The smaller particles were generally deposited near the higher elevations but not directly on top of the ridge.

The simulations all show an almost identical north-west flow with little variation among the images. Of interest are the apparent terrain effects in Figure 52 with respect

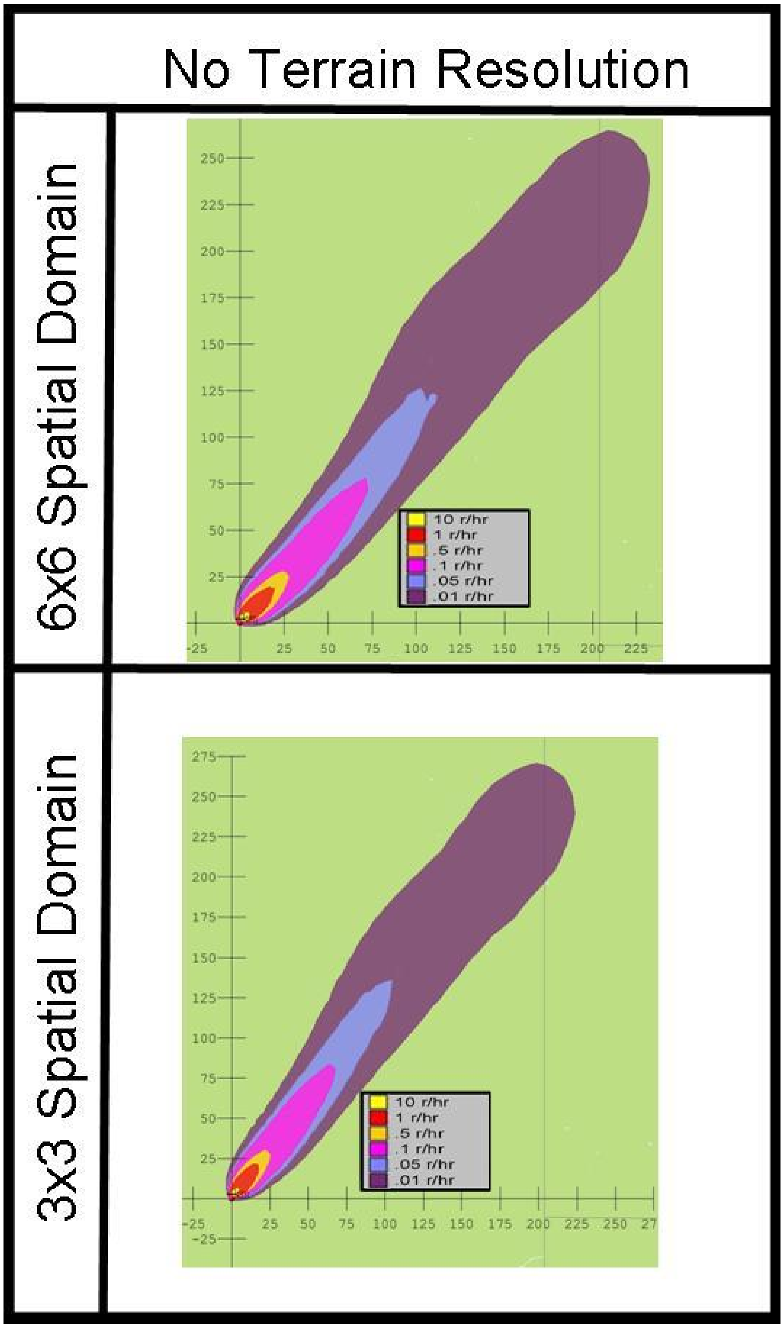


Figure 51 Johnie Boy Simulations Using No Terrain

to the small/35K simulation image. The tail end of the plume area widens to envelop two nearby mountaintops while simultaneously being impeded to the south by a third. Overall, the simulations' small directional deviation from the observed contour plots results in large disparities between observed and modeled contour locations.

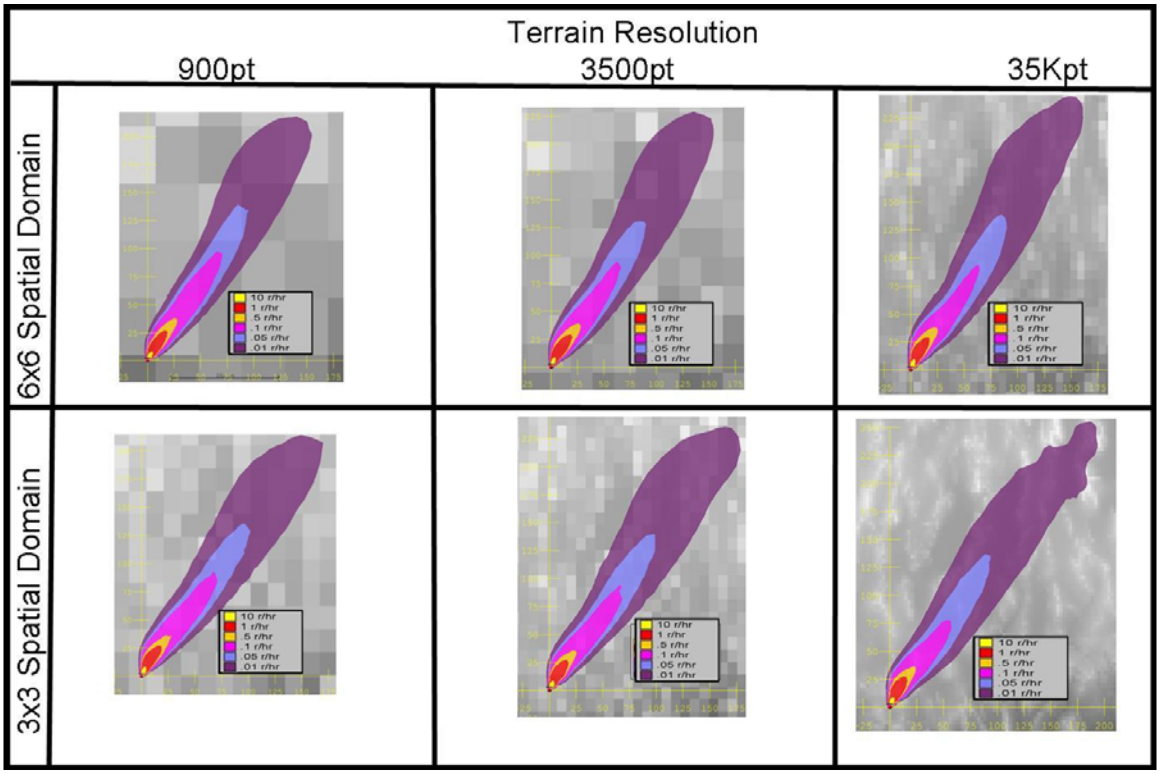


Figure 52 Johnie Boy Simulations Using Terrain

Though not quite as severe as the Smoky analysis, the Johnie Boy NAD and MOE value graphs are of limited practical use due to the absolute differences between values. However, of the main effects, the domain size and terrain resolution effects seem to be of the least value while the contour level seems to contain enough variance to be of some possible value (See Figures 53, 54, and 55). The ANOVA does confirm that the domain size (.24 p-value for NAD) and terrain resolution (.22 p-value for NAD) effects have no significant impact on the accuracy of the predicted model. All three graphs are analyzed and characterized by grouping the contour levels into three distinct groups. The 10 r/hr contour level is determined to be the most accurate while the .05 and .1 r/hr contours are, as a group, the least accurate. Finally, the .01, .5, and 1 r/hr contour levels are found to have an accuracy level less than the 10 r/hr dose rate but better than the .05 and .1 r/hr group.

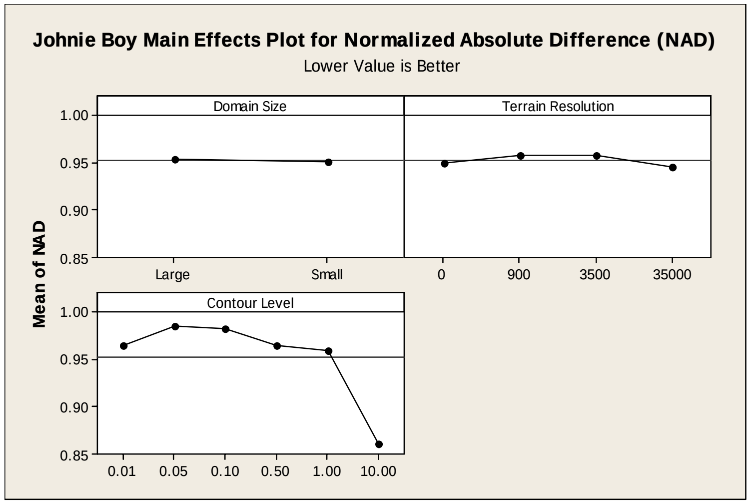


Figure 53 NAD vs. Main Effects (Johnie Boy)

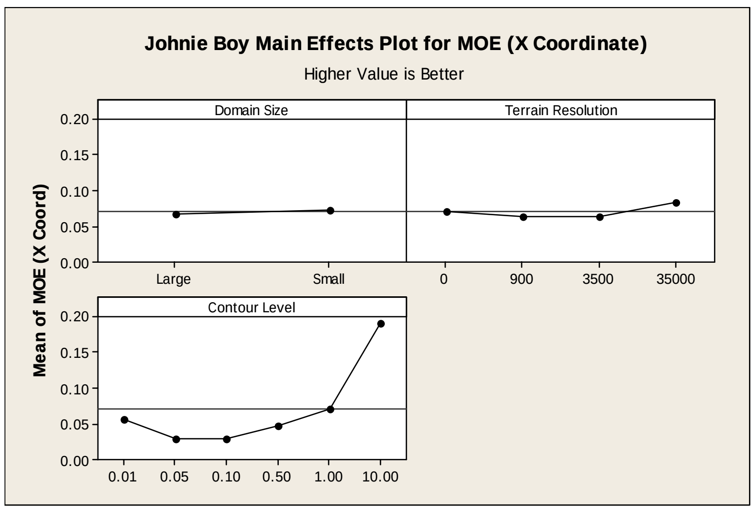


Figure 54 MOE x-coordinate vs. Main Effects (Johnie Boy)

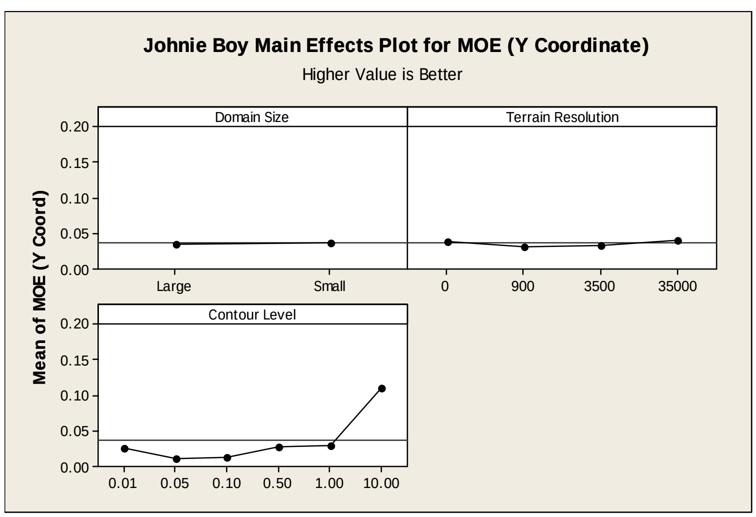


Figure 55 MOE y-coordinate vs. Main Effects (Johnie Boy)

Tests George, Ess, and Zucchini (Grouped)

The George, Ess, and Zucchini simulation results are grouped together in order to possibly glean additional information. The intent of this grouping is to increase the statistical population from eight (2 domain sizes by 4 terrain resolutions) to 24 (3 tests by 2 domain sizes by 4 terrain resolutions). The premise is that a larger sample will lead to a clearer understanding of the entire population. Considering that the entire population of nuclear tests that caused local fallout is no more than 300 test detonations [1], this grouping represents at least 1% of the entire population for fallout-producing nuclear tests conducted by the U.S.

Visually, the domain size seems to be of little consequence in terms of accuracy while running simulations with terrain probably increases accuracy. However, it seems

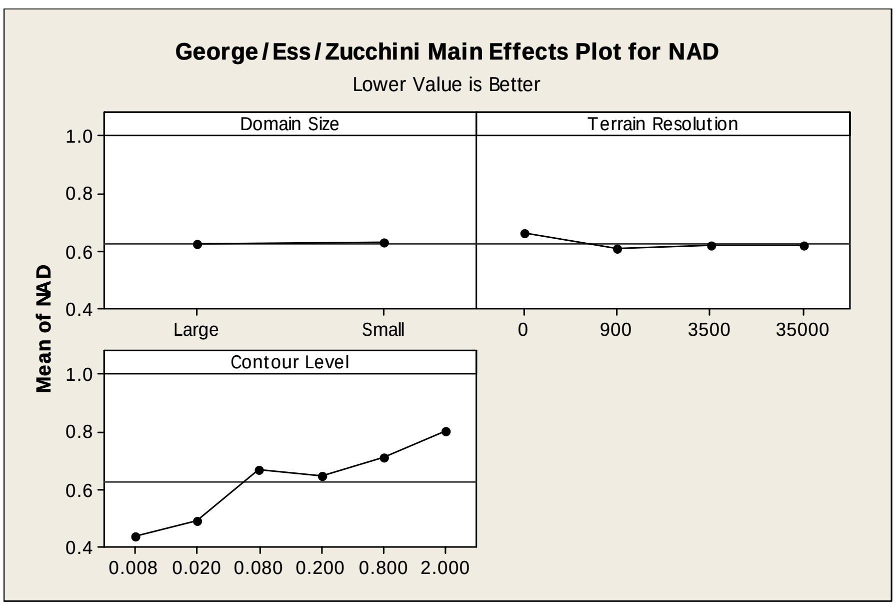


Figure 56 NAD vs. Main Effects (George, Ess, and Zucchini)

that using terrain resolutions beyond 900 points can possibly diminish this accuracy gain. The graphs further indicate a general increase in model accuracy as lower contour levels are considered, that is to say, lower dose-rate contours are more accurate than higher dose-rate contours.

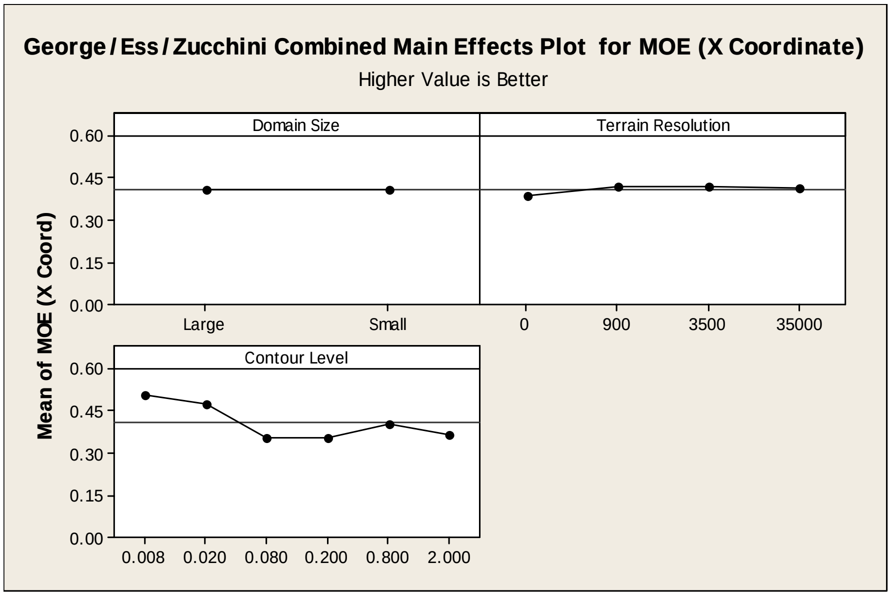


Figure 57 MOE x-coordinate vs. Main Effects (George, Ess, and Zucchini)

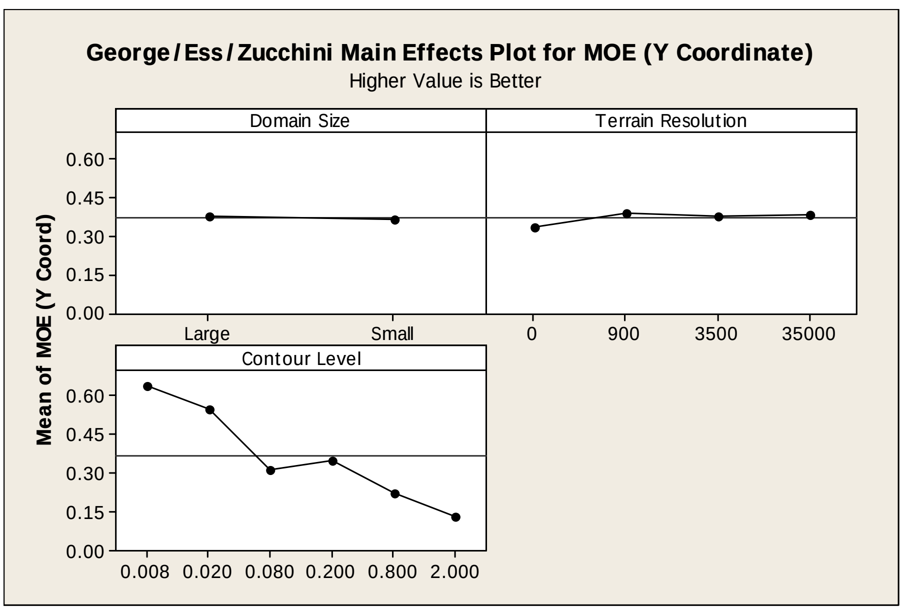


Figure 58 MOE y-coordinate vs. Main Effects (George, Ess, and Zucchini)

The analysis does confirm that fact that domain size does not statistically enhance the accuracy of any given simulation (.85 p-value for NAD). Contrary to visual observation, the ANOVA reveals that terrain resolution also plays no significant role in simulation accuracy (.82 p-value for NAD). Though the MOE x-coordinate shows no differentiation between contour levels for accuracy improvement, the NAD and MOE y-coordinate graphs do show some variation. The contour levels for the NAD are divided into two groups. The .008 and .02 r/hr contour levels were shown to have a better accuracy than the remaining contour levels. The MOE y-coordinate graph shows a three-way division. The .008 and .02 r/hr contour levels were again grouped and found to be the most accurate followed by the .08 and .2 r/hr contour levels. Finally, the .8 and 2 r/hr contour levels are found to be the least accurate of the three groups.

All Six Tests (Grouped)

All tests are grouped in order to increase the sample population from three to six. Doing so, however, prevents the analysis of contour levels as a main effect due to the Priscilla, Smoky, and Johnie Boy fallout data having different contour level descriptions in the DASA-EX document than the George, Ess, and Zucchini tests. While the contour level effect is removed from the analysis, the domain size and terrain resolution effects are more statistically valid due to the increased sample size.

Figures 59, 60, and 61 all show a probable insignificant accuracy effect based on domain size and terrain resolution. For this group of tests the effect of “test” is calculated. Though the tests are listed in chronological order and values are connected in a trend-like manner, no visual trends are considered. The test effect is merely graphed to make singular observations about how tests fared, in general, with respect to the NAD and MOE coordinate accuracy.

The ANOVA confirms the visual observation that neither domain size (.73 p-value for NAD) nor terrain resolution (.63 p-value for NAD) has a verifiable effect on modeling accuracy. The test effect shows that the Ess simulations are the most accurate of the six tests studied in this research.

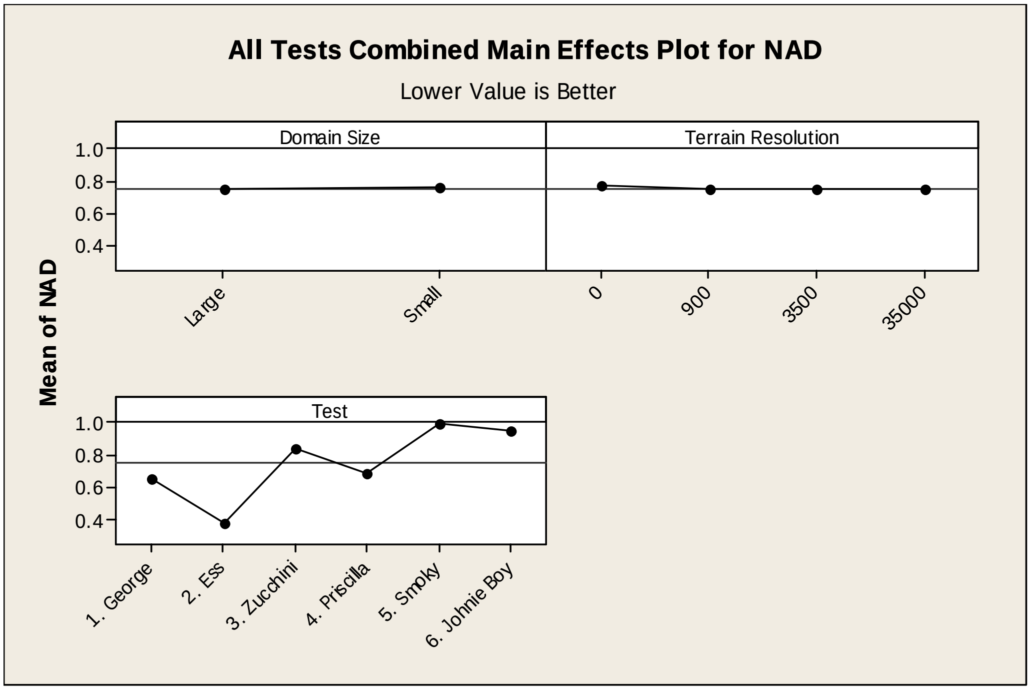


Figure 59 NAD vs. Main Effects (All Detonations)

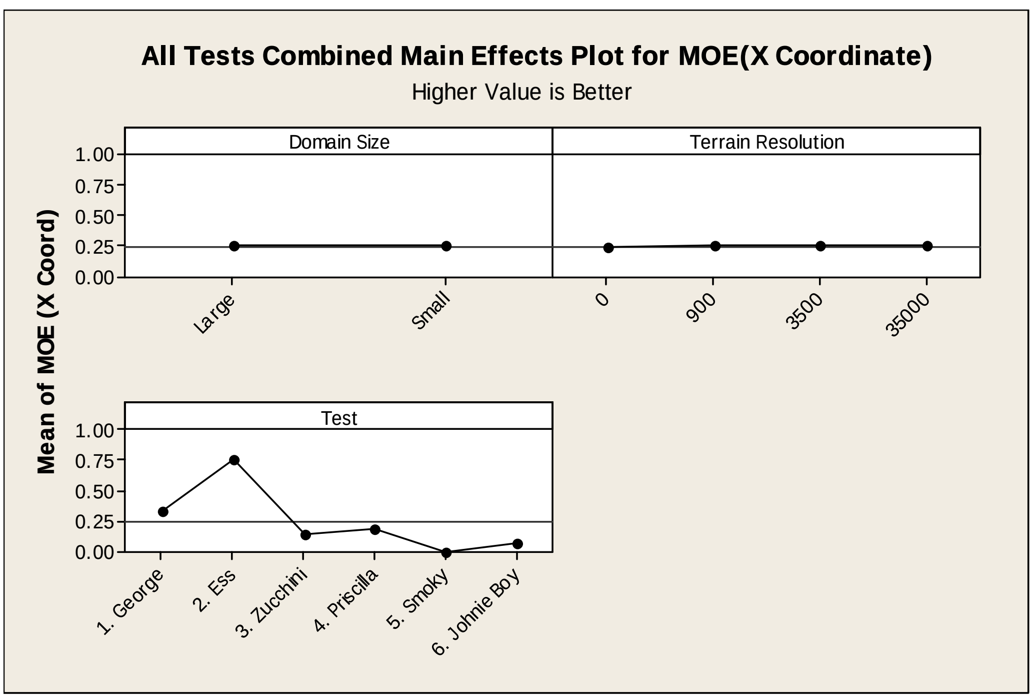


Figure 60 MOE x-coordinate vs. Main Effects (All Detonations)

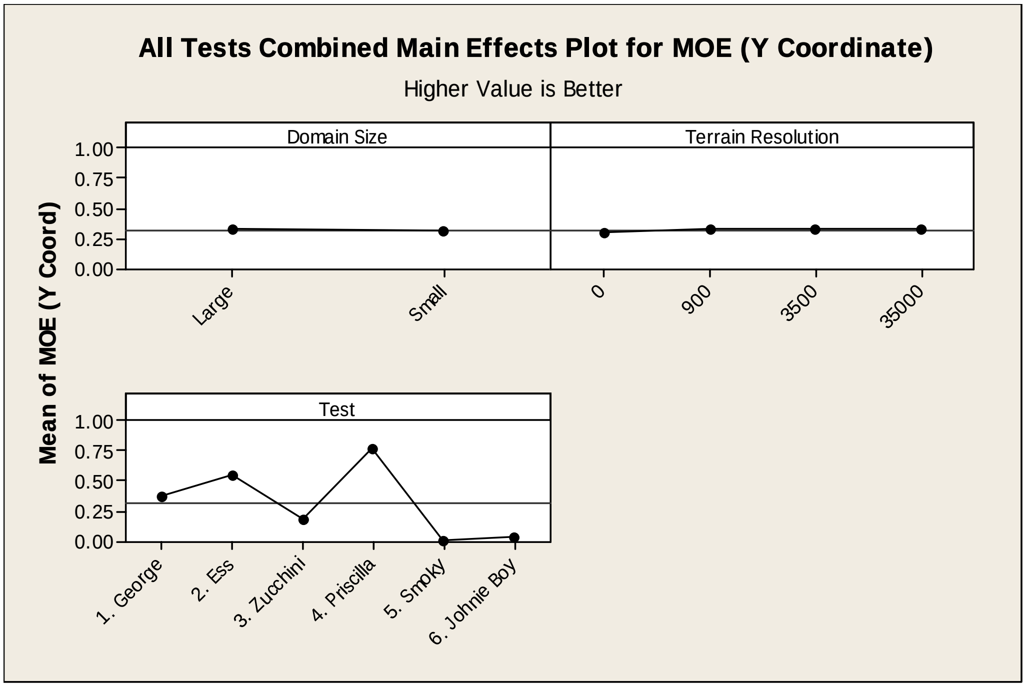


Figure 61 MOE y-coordinate vs. Main Effects (All Detonations)

V. Summary and Conclusions

Chapter Overview

This chapter summarizes the analysis performed in chapter four and identifies the possible emerging trends. It also includes a set of broad conclusion statements regarding this work. Finally, recommendations are made for future research regarding this topic.

Summary

In general, every effect shows some accuracy differentiation in this research. Though only the George and Zucchini simulations show accuracy differentiation with regards to domain size, all simulations do show an accuracy effect when taking contour levels into consideration. The y-coordinate of the MOE is positively affected as the George, Ess, and Priscilla simulations show increasing accuracy by using any of the non-zero terrain resolutions. The Smoky models show a marked detriment in accuracy when using the 35K terrain resolution while the Zucchini simulations show an improvement when using no terrain. The George and Zucchini simulations show that the MOE x-coordinate is affected while the other tests' simulations are unaffected. The NAD shows affect in the George, Ess, and Zucchini simulations in the same manner as the MOE y-coordinate. Unfortunately, all of the statistical significance due to terrain resolution is lost when tests are grouped for analysis. This is due to some MOE and NAD values being very closely grouped in terms of absolute numbers. The implication for this loss of statistical significance is that test simulations must be viewed individually in order to observe statistical trends.

The contour-level effect shows accuracy differentiation in every test. Due to the different contour levels used in many of these tests, contour levels are referred to in relative terms. Low, mid, and high refer to a contour level's relative placement for a given test. Both the George and

Zucchini simulations show that, in general, accuracy improves when considering lower and lower contour levels. The Priscilla simulations also show this apparent trend with regard to the NAD and MOE x-coordinate. The MOE y-coordinate for Priscilla models favors mid contour levels. In contrast, the Smoky simulations show that, were it not for the .1 r/hr contour line, high contour levels are more accurate than mid or low contour levels. This is the case for all main effects. The Ess simulations show NAD and MOE y-coordinate accuracy improvement when mid contour levels are considered, however, the MOE x-coordinate shows best results when high contour levels are studied. Johnie Boy simulations show higher accuracy with high contour levels and lowest accuracy with mid contour levels. This is true for all main effects. When the three-way grouping is studied, the NAD and MOE y-coordinate show accuracy improvements when using the lowest contour levels.

Conclusions

Though the Zucchini simulations show NAD accuracy improvements when using a larger domain, the overwhelming evidence is that domain size plays no part in modeling accuracy.

There is also little support for a terrain resolution effect when considering the three- and six-way groupings only. However, looking at the George, Ess, and Priscilla simulations, it is apparent that these simulations use weather data that is much closer to actual test-day weather than any other of the detonations studied. Moreover, the contour plots of the other three tests studied vary enough from the observed data that it is clear that the reanalysis weather data is not good enough to produce statistically identifiable effects due to terrain. Therefore, I conclude that terrain resolution has a significant, positive effect on model accuracy if weather data is sufficiently similar to actual test-day weather.

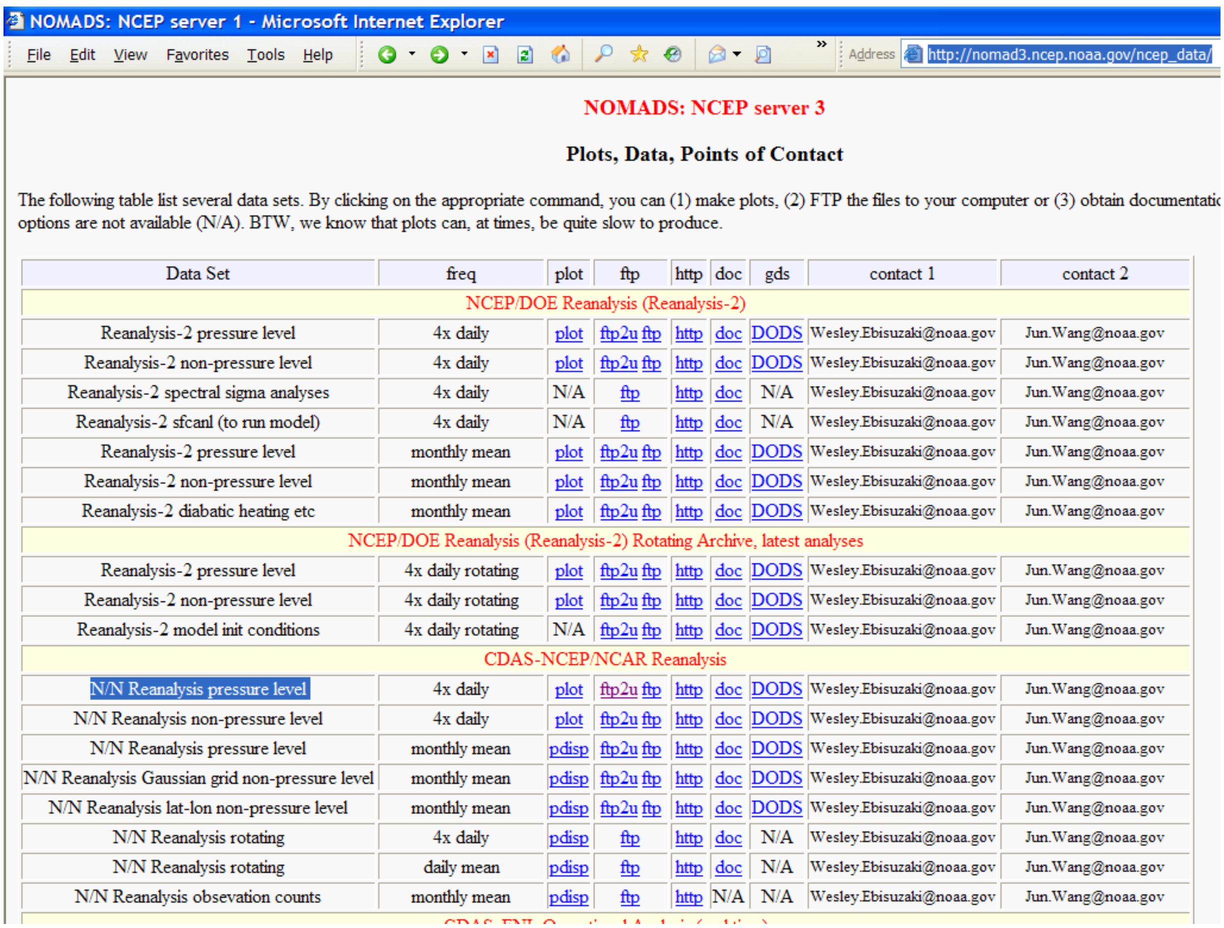
The contour level effect is much more apparent than the terrain resolution effect in that there are two conclusions that can be drawn. The first is that the more inaccurate the simulation, the more likely it is that the high contour levels will be less inaccurate than the low contour levels. This is probably due to the fact that the larger particles, which are responsible for a major portion of the higher contour levels, are less affected by inaccurate winds than are the smaller particles which make up the relatively large contour areas. Conversely, the second conclusion is that given a more accurate weather field, the lower contour levels will tend to be more accurate.

Future Research Directions

Given the conclusions of this thesis there are three areas of future research that could be of significant value in making HPAC a more viable tool for the emergency planner and/or first responders. These areas of research are listed in order of perceived value. First, this research can be repeated using simulation weather data that incorporates historical local weather observations into the reanalysis weather file. This incorporated data would allow HPAC simulations to execute using weather that has a finer spatial and/or temporal resolution. Along this same idea is accessing the reanalysis weather database and building an HPAC weather file using all local observations. Second, this work can be replicated using tests in which the historical observed wind fields are similar to reanalysis winds. Though this is essentially ‘hand picking the jury’, it would confirm or deny the conclusion that increasing terrain resolution leads to a more accurate model if and only if the weather data used for modeling is of sufficient accuracy. The third area of research is to extract terrain data out of HPAC. This would allow the comparisons of true topographical areas as opposed to two-dimensionally rendered areas in which the z-component of topography is ignored.

Appendix A: Reanalysis Data Acquisition

Go to http://nomad3.ncep.noaa.gov/ncep_data/. Find the “CDAS-NCEP/NCAR Reanalysis” section, specifically the “N/N Reanalysis pressure level 4x daily” subsection. Click on the “ftp2u” link located in that subsection. A partial view of the website with the proper line highlighted can be seen below.


NOMADS: NCEP server 1 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://nomad3.ncep.noaa.gov/ncep_data/

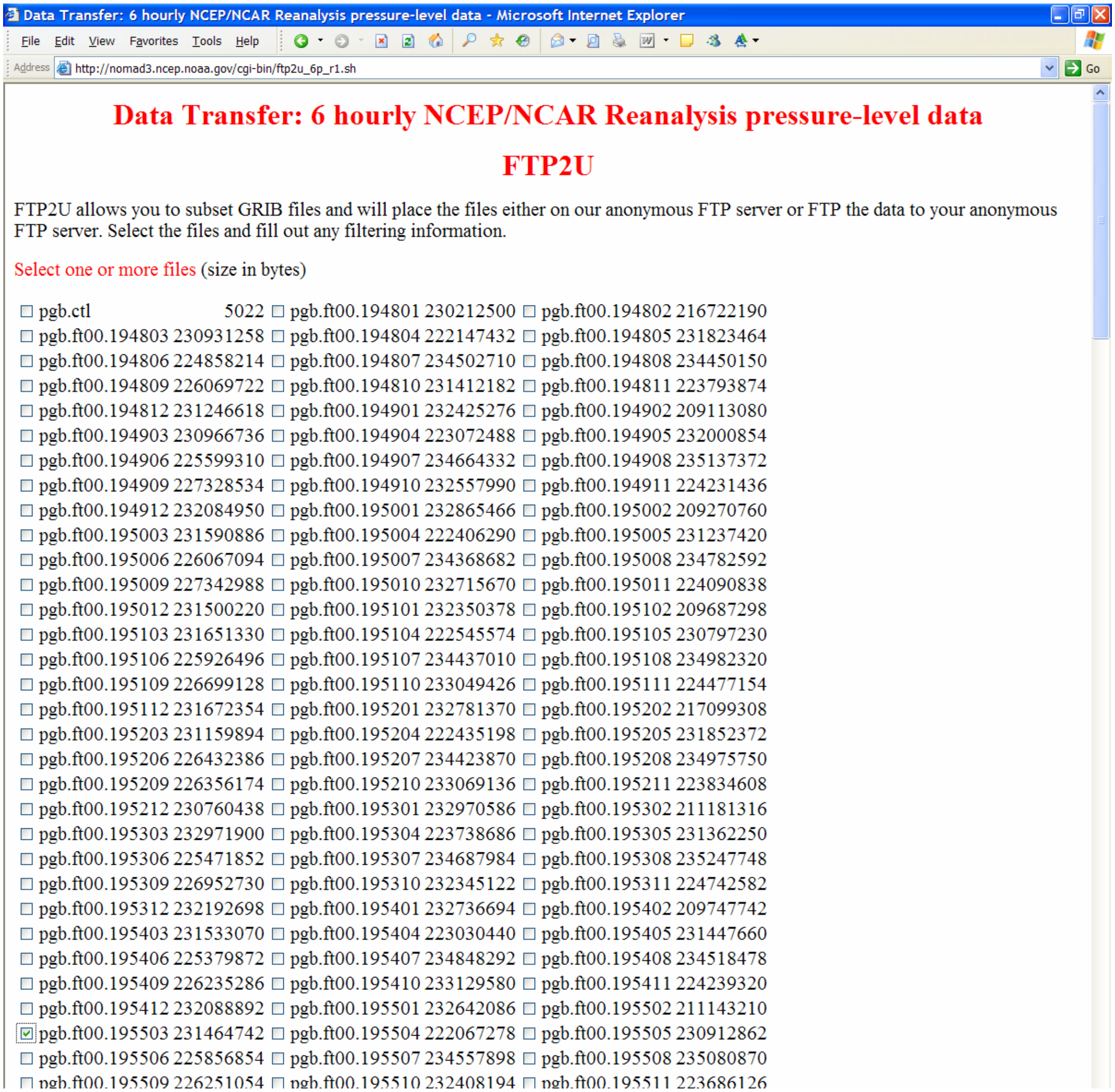
NOMADS: NCEP server 3

Plots, Data, Points of Contact

The following table list several data sets. By clicking on the appropriate command, you can (1) make plots, (2) FTP the files to your computer or (3) obtain documentatic options are not available (N/A). BTW, we know that plots can, at times, be quite slow to produce.

Data Set	freq	plot	ftp	http	doc	gds	contact 1	contact 2
NCEP/DOE Reanalysis (Reanalysis-2)								
Reanalysis-2 pressure level	4x daily	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 non-pressure level	4x daily	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 spectral sigma analyses	4x daily	N/A	ftp	http	doc	N/A	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 sfcanl (to run model)	4x daily	N/A	ftp	http	doc	N/A	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 pressure level	monthly mean	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 non-pressure level	monthly mean	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 diabatic heating etc	monthly mean	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
NCEP/DOE Reanalysis (Reanalysis-2) Rotating Archive, latest analyses								
Reanalysis-2 pressure level	4x daily rotating	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 non-pressure level	4x daily rotating	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
Reanalysis-2 model init conditions	4x daily rotating	N/A	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
CDAS-NCEP/NCAR Reanalysis								
N/N Reanalysis pressure level	4x daily	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis non-pressure level	4x daily	plot	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis pressure level	monthly mean	pdisp	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis Gaussian grid non-pressure level	monthly mean	pdisp	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis lat-lon non-pressure level	monthly mean	pdisp	ftp2u ftp	http	doc	DODS	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis rotating	4x daily	pdisp	ftp	http	doc	N/A	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis rotating	daily mean	pdisp	ftp	http	doc	N/A	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov
N/N Reanalysis observation counts	monthly mean	pdisp	ftp	http	N/A	N/A	Wesley.Ebisuzaki@noaa.gov	Jun.Wang@noaa.gov

Clicking on the proper link will take you to http://nomad3.ncep.noaa.gov/cgi-bin/ftp2u_6p_r1.sh. This web address could be typed in directly. The top of this website lists all of the months for which reanalysis data is available. At the time of this research, the dates available spanned from January 1948 thorough December 2005. Select a month (or months) for which reanalysis weather data is required. As an example, the included website pictures will document downloading 23 March 1955 through 27 March 1955 for the large spatial domain used in this document.



Data Transfer: 6 hourly NCEP/NCAR Reanalysis pressure-level data

FTP2U

FTP2U allows you to subset GRIB files and will place the files either on our anonymous FTP server or FTP the data to your anonymous FTP server. Select the files and fill out any filtering information.

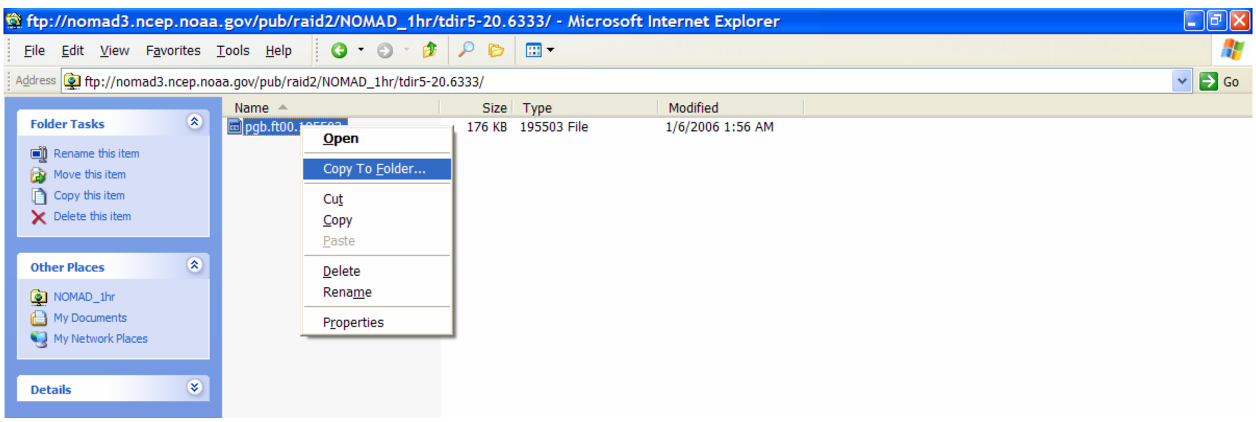
Select one or more files (size in bytes)

☐ pgb.ctl	5022	☐ pgb.ft00.194801	230212500	☐ pgb.ft00.194802	216722190
☐ pgb.ft00.194803	230931258	☐ pgb.ft00.194804	222147432	☐ pgb.ft00.194805	231823464
☐ pgb.ft00.194806	224858214	☐ pgb.ft00.194807	234502710	☐ pgb.ft00.194808	234450150
☐ pgb.ft00.194809	226069722	☐ pgb.ft00.194810	231412182	☐ pgb.ft00.194811	223793874
☐ pgb.ft00.194812	231246618	☐ pgb.ft00.194901	232425276	☐ pgb.ft00.194902	209113080
☐ pgb.ft00.194903	230966736	☐ pgb.ft00.194904	223072488	☐ pgb.ft00.194905	232000854
☐ pgb.ft00.194906	225599310	☐ pgb.ft00.194907	234664332	☐ pgb.ft00.194908	235137372
☐ pgb.ft00.194909	227328534	☐ pgb.ft00.194910	232557990	☐ pgb.ft00.194911	224231436
☐ pgb.ft00.194912	232084950	☐ pgb.ft00.195001	232865466	☐ pgb.ft00.195002	209270760
☐ pgb.ft00.195003	231590886	☐ pgb.ft00.195004	222406290	☐ pgb.ft00.195005	231237420
☐ pgb.ft00.195006	226067094	☐ pgb.ft00.195007	234368682	☐ pgb.ft00.195008	234782592
☐ pgb.ft00.195009	227342988	☐ pgb.ft00.195010	232715670	☐ pgb.ft00.195011	224090838
☐ pgb.ft00.195012	231500220	☐ pgb.ft00.195101	232350378	☐ pgb.ft00.195102	209687298
☐ pgb.ft00.195103	231651330	☐ pgb.ft00.195104	222545574	☐ pgb.ft00.195105	230797230
☐ pgb.ft00.195106	225926496	☐ pgb.ft00.195107	234437010	☐ pgb.ft00.195108	234982320
☐ pgb.ft00.195109	226699128	☐ pgb.ft00.195110	233049426	☐ pgb.ft00.195111	224477154
☐ pgb.ft00.195112	231672354	☐ pgb.ft00.195201	232781370	☐ pgb.ft00.195202	217099308
☐ pgb.ft00.195203	231159894	☐ pgb.ft00.195204	222435198	☐ pgb.ft00.195205	231852372
☐ pgb.ft00.195206	226432386	☐ pgb.ft00.195207	234423870	☐ pgb.ft00.195208	234975750
☐ pgb.ft00.195209	226356174	☐ pgb.ft00.195210	233069136	☐ pgb.ft00.195211	223834608
☐ pgb.ft00.195212	230760438	☐ pgb.ft00.195301	232970586	☐ pgb.ft00.195302	211181316
☐ pgb.ft00.195303	232971900	☐ pgb.ft00.195304	223738686	☐ pgb.ft00.195305	231362250
☐ pgb.ft00.195306	225471852	☐ pgb.ft00.195307	234687984	☐ pgb.ft00.195308	235247748
☐ pgb.ft00.195309	226952730	☐ pgb.ft00.195310	232345122	☐ pgb.ft00.195311	224742582
☐ pgb.ft00.195312	232192698	☐ pgb.ft00.195401	232736694	☐ pgb.ft00.195402	209747742
☐ pgb.ft00.195403	231533070	☐ pgb.ft00.195404	223030440	☐ pgb.ft00.195405	231447660
☐ pgb.ft00.195406	225379872	☐ pgb.ft00.195407	234848292	☐ pgb.ft00.195408	234518478
☐ pgb.ft00.195409	226235286	☐ pgb.ft00.195410	233129580	☐ pgb.ft00.195411	224239320
☐ pgb.ft00.195412	232088892	☐ pgb.ft00.195501	232642086	☐ pgb.ft00.195502	211143210
☑ pgb.ft00.195503	231464742	☐ pgb.ft00.195504	222067278	☐ pgb.ft00.195505	230912862
☐ pgb.ft00.195506	225856854	☐ pgb.ft00.195507	234557898	☐ pgb.ft00.195508	235080870
☐ pgb.ft00.195509	226251054	☐ pgb.ft00.195510	232408194	☐ pgb.ft00.195511	223686126

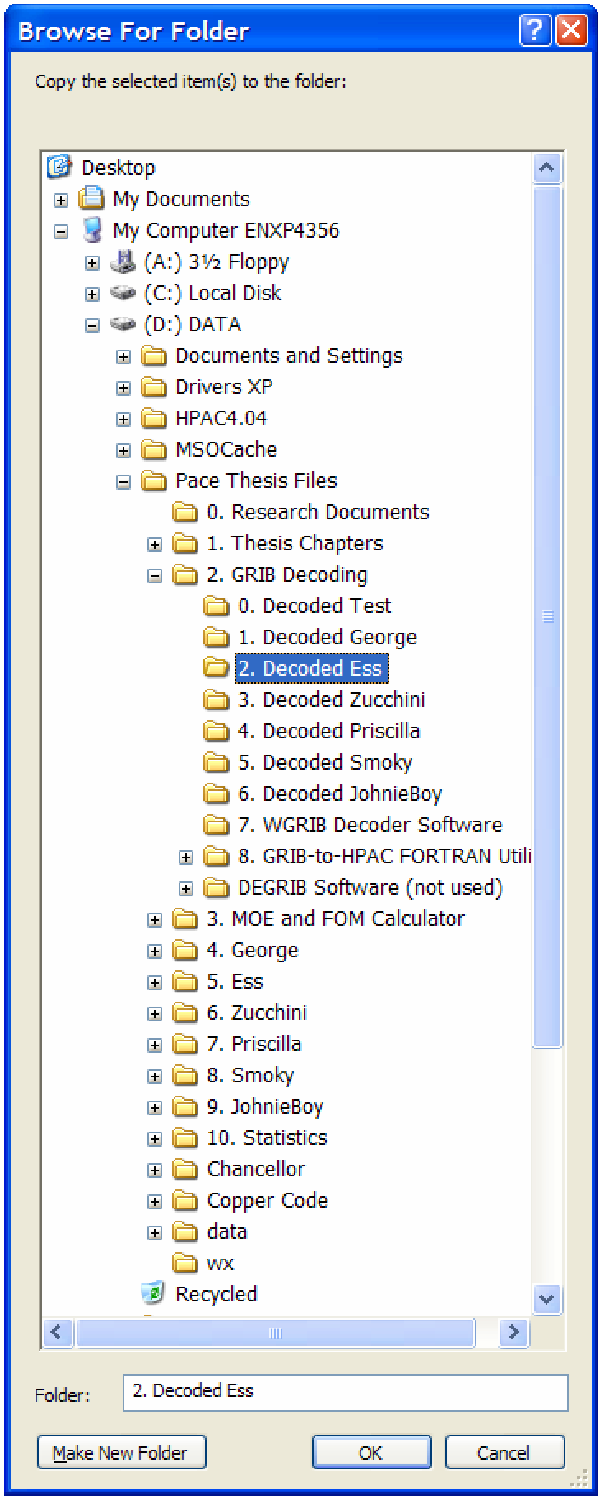
After this has been accomplished, scroll down the web page until you reach the "Grib Filter" section. At this point we scale back the file size (written to the right of each month in the above picture) to a more reasonable size. See below for an example of weather data downloaded for this research.

At this point the “Start Download” button is clicked and the user is taken to a web site resembling:

To download the reanalysis weather information the user can click the middle link. This will direct the user to a directory with the requested file as its only contents.



The user then right clicks the file and chooses "Copy To Folder…" Upon choosing this, the user can specify where he/she wants to store the grib file.

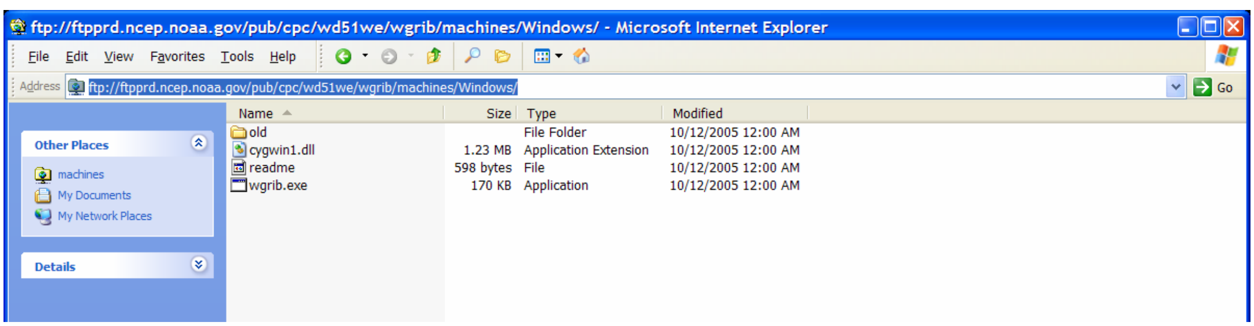

Browse For Folder
Copy the selected item(s) to the folder:
Desktop
My Documents
My Computer ENXP4356
(A:) 3½ Floppy
(C:) Local Disk
(D:) DATA
Documents and Settings
Drivers XP
HPAC4.04
MSOCache
Pace Thesis Files
0. Research Documents
1. Thesis Chapters
2. GRIB Decoding
0. Decoded Test
1. Decoded George
2. Decoded Ess
3. Decoded Zucchini
4. Decoded Priscilla
5. Decoded Smoky
6. Decoded JohnieBoy
7. WGRIB Decoder Software
8. GRIB-to-HPAC FORTRAN Utili
DEGRIB Software (not used)
3. MOE and FOM Calculator
4. George
5. Ess
6. Zucchini
7. Priscilla
8. Smoky
9. JohnieBoy
10. Statistics
Chancellor
Copper Code
data
wx
Recycled
Folder:
2. Decoded Ess
Make New Folder
OK
Cancel

Appendix B: Using WGRIB Software

WGRIB software is a shareware utility program that can decompress grib files into two separate text files. These test files must be used in conjuction with each other as outlined in Chapter 3 of this document. This appendix will explain to to obtain and properly use this software in such a way that the weather data can be transformed into an HPAC compatible file by using the FORTRAN code in Appendix C.

This author obtained the software from the web address http://www.cpc.ncep.noaa.gov/products/wesley/wgrib.html. This site offers a download option. The web address ftp://ftpprd.ncep.noaa.gov/pub/cpc/wd51we/wgrib/machines/Windows/ is a direct link to the ftp library where the windows version of the software is stored. Though this is the Windows version of the software, the software must be run from a command prompt.

There are two files that must be downloaded and placed in the same directory. They are the wgrib.exe and cygwin1.dll files.



Once downloaded, a command prompt must be opened and set to the directory where the wgrib software and grib files are located. Below is a sample session using wgrib software.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS>d:

D:\>cd Pace Thesis Files
```

```
D:\Pace Thesis Files>cd 2. Grib decoding

D:\Pace Thesis Files\2. GRIB Decoding>cd 7. WGRIB decoder software

D:\Pace Thesis Files\2. GRIB Decoding\7. WGRIB Decoder Software>dir
 Volume in drive D is DATA
 Volume Serial Number is C8A9-F88C

 Directory of D:\Pace Thesis Files\2. GRIB Decoding\7. WGRIB Decoder Software

01/05/2006  09:46 PM    <DIR>          .
01/05/2006  09:46 PM    <DIR>          ..
10/12/2005  07:14 PM         1,295,582 cygwin1.dll
10/20/2005  05:19 PM           251,352 sample.grb
12/09/2005  12:02 PM            42,496 WGRIB Documentation.doc
10/12/2005  07:16 PM               598 WGRIB readme.html
10/12/2005  07:14 PM           174,329 wgrib.exe
               5 File(s)      1,764,357 bytes
               2 Dir(s)  95,697,764,352 bytes free

D:\Pace Thesis Files\2. GRIB Decoding\7. WGRIB Decoder Software>wgrib sample.grb
 -V -d all -text -o WxData.txt -> WxDataDecoder.txt
argument: -V ????
argument: -d ????
argument: all ????
argument: -text ????
argument: -o ????
argument: WxData.txt ????
argument: - ????

D:\Pace Thesis Files\2. GRIB Decoding\7. WGRIB Decoder Software>dir
 Volume in drive D is DATA
 Volume Serial Number is C8A9-F88C

 Directory of D:\Pace Thesis Files\2. GRIB Decoding\7. WGRIB Decoder Software

01/05/2006  09:49 PM    <DIR>          .
01/05/2006  09:49 PM    <DIR>          ..
10/12/2005  07:14 PM         1,295,582 cygwin1.dll
10/20/2005  05:19 PM           251,352 sample.grb
12/09/2005  12:02 PM            42,496 WGRIB Documentation.doc
10/12/2005  07:16 PM               598 WGRIB readme.html
10/12/2005  07:14 PM           174,329 wgrib.exe
01/05/2006  09:49 PM           208,634 WxDataDecoder.txt
               6 File(s)      1,972,991 bytes
               2 Dir(s)  95,697,555,456 bytes free
```

D:\Pace Thesis Files\2. GRIB Decoding\7. WGRIB Decoder Software>

Notice that the resultant files are stored in the same directory as the wgrib software. These are standard text files that require no modification before using the FORTRAN utility in Appendix C.

The command used flags that can be identified below.

Inventory/diagnostic-output selections		
-s/-v	short/verbose inventory	
-V	diagnostic output (not inventory)	
(none)	regular inventory	
Options		
-PDS/-PDS10	print PDS in hex/decimal	
-GDS/-GDS10	print GDS in hex/decimal	
-verf	print forecast verification time	
-ncep_opn/-ncep_rean	default T62 NCEP grib table	
-4yr	print year using 4 digits	
-min	print minutes	
-ncep_ens	ensemble info encoded in ncep format	
Decoding GRIB selection		
-d [record number|all]	decode record number	
-p [byte position]	decode record at byte position	
-i	decode controlled by stdin (inventory list)	
(none)	no decoding	
Options		
-text/-ieee/-grib/-bin	convert to text/ieee/grib/bin (default)	
-nh/-h	output will have no headers/headers (default)	
-dwdgrib	output dwd headers, grib (do not append)	
-H	output will include PDS and GDS (-bin/-ieee only)	
-append	append to output file	
-o [file]	output file name, 'dump' is default	

Appendix C: WGRIB-2-PRF FORTRAN Utility

This appendix contains the FORTRAN source code for a utility that transforms the two resultant text files from using the wgrib software into a readily usable HPAC weather profile file. This is in no way meant to be polished in any way. Many remarked out lines of code were used in debugging and were left in case modifications were made that required more debugging. The source code is broken down into a main program, seven modules, and one data file.; GRIB2HPAC.f90, Kinds.f90, Global.f90, LocationArray.f90, TimeArray.f90, LevelArray.f90, WxArray.f90, HPAC_PRF_WRITER.f90, and Surface Elevation.dat

GRIB2HPAC.f90

```
Program GRIB2HPAC
!*************************************************************************************************
!       Purpose
!
!       This program will take the 2 files decoded by wgrib.exe and rearrange the data in
!          an HPAC .prf file.  One of the files is the inventory file that describes the data in
!          the data file and the other is the file containing the actual weather data.
!
!       Date                  Programmers           Description of Change
!       ====                  ==========            =====================
!       24 OCT 05             MAJ Kevin Pace Original Code
!*************************************************************************************************
Use Kinds
Use Globals
Use LocationTools
Use TimeTools
Use LevelTools
Use WxTools
Use PrfWriter

Implicit None

Integer,       Allocatable:: DTG(:)   ! Date-Time-Groups in which Wx data is avail [YYYYMMDDHH]
Integer,       Allocatable:: Level(:) ! Pressure levels for which Wx data is avail [mb]
Type(Loc),     Allocatable:: Location(:) ! Locations for which Wx data is avail [Lon, Lat]
Type(WxPoint), Allocatable:: WXPT(:,:,:) ! 3D array (Level, Location, Time) of Wx data points


!*************************************************************************************************
! Get Filenames of WGRIB-Decoded Inventory and Data Files
Write(*,*) 'Enter filename of inventory file that was decoded by WGRIB: '
Read(*,*) Inventory
Write(*,*) 'Enter filename of matching data file that was also decoded by WGRIB: '
Write(*,*) 'It is imperative that the two files were created by a single WGRIB decoding'
Read(*,*) DataFile

!*************************************************************************************************
! Allocate and Initialize Location Array
Call GetLocationSize (Inventory)                   !Find the number of reanalysis points
                                                   !in this file.  Also returns flags for
                                                   !separating data in data file eg "6 6"
Allocate (Location(1:LocSize))        !Allocate the array
Location%Lat = -9999.0_dp             !Initialize the Array
Location%Lon = -9999.0_dp

Write(*,*) "There are ", LocSize, " locations covered in this file"
Write(*,*)

!*************************************************************************************************
```

```
! Fill Location Array with all lat/lon locations for which reanalysis data is available
Call FillLocation (Location, Inventory)

Write(*,*) "Location Number         Longitude              Latitude"
Do i = 1, LocSize
Write(*, 1000) i, Location(i)%Lon, Location(i)%Lat
1000 Format (I4, 16x, F9.2,13x,F9.2)
End Do

Write(*,*)

!*************************************************************************************************
! Allocate and Initialize DTG Array
Call GetTimeSize (TimeSize, Inventory)  !Also counts the number of records in the reanalysis file
Allocate (DTG(1:TimeSize))
DTG = -9999._dp                          !Initialize character array

Write(*,*) "Number of Records:  ", Rec
Write(*,*) "There are ", TimeSize ,"DTGs that this file covers"
!*************************************************************************************************
! Fill Array with DTGs that the reanalysis data covers.  Early -> Late (Just as Inventory File)
Call FillTimeArray (DTG, Inventory)

Write(*,*) DTG
Write(*,*)

!*************************************************************************************************
! Allocate Layer Array
Call GetLevelSize (LevelSize, Inventory)
Allocate (Level(1:LevelSize))
Level = -9999.0_dp

! Fill Layer Array with Pressure Levels
Call FillLevelArray(Level, Inventory, LevelSize)

Write(*,*) Level
Write(*,*)
!*************************************************************************************************
! Allocate and Initialize WXPT Array (Array of TYPE: WxPoint)
Allocate (WXPT(1:LevelSize, 1:LocSize, 1:TimeSize))

WXPT%HGT    = -9999.0_dp                                    !Initialize the Array
WXPT%T      = -9999.0_dp
WXPT%U      = -9999.0_dp
WXPT%V      = -9999.0_dp
WXPT%RH     = -9999.0_dp
WXPT%WndDir = -9999.0_dp
WXPT%WndSpd = -9999.0_dp

! Fill WXPT Array with data from data file from WGRIB
Call FillWXPTArray (WXPT, Level, Location, DTG, Inventory, DataFile)

!Debugger that writes a duplicate of the datafile.  I imported both files into Excel
!and compared them for line length and value-to-value matching

!OPEN (UNIT = 40, FILE="OutputTest.txt", STATUS='OLD', ACTION='WRITE', IOSTAT=ierror1)
!If (ierror1 .NE. 0) Write(*,*) 'Cant Open this file'

Call WritePRF (WxPT, DTG, Location, Level, Inventory)

End Program GRIB2HPAC
```

Kinds.f90

```
Module Kinds

       Implicit None
       Public

!*************************************************************************************************
!      Date                  Programmers                           Description of Change
!      ====                  ==========                            =====================
!      24 Jan 05             MAJ Kevin Pace                  Original Code
!*************************************************************************************************

       Integer,Parameter:: sp = Selected_Real_Kind(p=6)
       Integer,Parameter:: dp = Selected_Real_Kind(p=14)

End Module Kinds
```

Global.f90

```
Module Globals

Use Kinds
Implicit None

Integer:: ierror1, ierror2                          ! Error Flag
Integer:: i, j, k                                           ! Loop Counters
Character(len=20):: Inventory, DataFile      ! Filenames for the 2 files decoded by wgrib.exe
Integer:: rec        ! Number of records in the Inventory (sets of data in data file)
Integer:: LonGrid, LatGrid, LocSize, TimeSize, LevelSize    ! Computed for the allocation of
arrays

Type :: WxPoint               ! Contains Wx values for a given location (lat/lon), press level,
and time
       Real(dp) :: HGT        ! Geopotential height at bottom of the layer [m]
       Real(dp) :: T  ! Air Temperature [K]
       Real(dp) :: U  ! E-W Wind Component (Wind is TO this direction; positive = East) [m/s]
       Real(dp) :: V  ! N-S Wind Component (Wind is TO this direction; positive = North) [m/s]
       Real(dp) :: RH ! Relative Humidity [%]
       Real(dp) :: WndDir     !Wind Azimuth (Clockwise from North; wind FROM this direction)
[unitless]
       Real(dp) :: WndSpd     !Wind Speed of the Wind Azimuth [m/s]
End Type WxPoint

Type :: Loc                   ! Location consists of a Latitude and Longitude
       Real(dp) :: Lat
       Real(dp) :: Lon
End Type Loc


End Module Globals
```

LocationArray.f90

```
Module LocationTools

!***************************************************************************************************
!      Computes the locations of all reanalysis weather data within the spatial boundary
!
!      Date                 Programmers                          Description of Change
!      ====                 ==========                           =====================
!      24 OCT 05            MAJ Kevin Pace                Original Code
!***************************************************************************************************

Use Kinds
Use Globals
Implicit None

Contains

Subroutine GetLocationSize (Inventory)
!***************************************************************************************************
! Extracts out # of lat/lon locations that the reanalysis file covers.  Should be HPAC spatial
! domain.  The first few lines of a typical reanalysis inventory file look like:
!
!rec 1:0:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=1000 levels=(3,232) grid=255 1000 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 7 66  num bits 6  BDS_Ref 7  DecScale 0 BinScale 0
!
!rec 2:92:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=925 levels=(3,157) grid=255 925 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 674 730  num bits 6  BDS_Ref 674  DecScale 0 BinScale 0
!
!**************************************************************************************************

Use Kinds
Use Globals
Implicit None

Character(Len=20), Intent(In):: Inventory    !Name of the reanalysis inventory file
Character(Len = 200) :: Line6        ! 6th Line of the Inventory File. Contains Grid numbers
```

```
Integer:: arrow       ! Pointer used to index my way across a line of text

ierror1 = 0

! Open File and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'error Opening Inventory File in GetLocSize Subroutine'

Do i = 1,5                                                            ! Move pointer over
the first 5 lines
       READ (20,*, IOSTAT = ierror1)
       If (ierror1 .NE. 0) EXIT
End Do

Read (20,'(a)', IOSTAT = ierror1) Line6             ! Read the 6th Line of the Inventory file
arrow = index(Line6,"(") + 1                        ! Find the ( before the Number of Lons
Read (Line6(arrow:),*) LonGrid                             ! Read the number of Lons in the grid
arrow = index(Line6,"x") + 1                        ! Find the x before the Number of Lats
Read (Line6(arrow:),*) LatGrid                             ! Read the number of Lats in the grid

LocSize = LonGrid * LatGrid

Close (20)

End Subroutine GetLocationSize


Subroutine FillLocation (Location, Inventory)
!************************************************************************************************
!Fills the Location array with Lats/Lons in the order that the data file lists values.
!For reanalysis files, the first value listed is for the most NW location.  After that it moves
!across the Northern-most lat in an Eastward direction.  When it runs to the the most NE location
!it starts at the second most northern lat and the most western lon reading across in an Easterly
!direction.  It continues this 'typewriter' approach of assigning values when it reaches the most
!SE location.
!************************************************************************************************

Use Kinds
Use Globals
Implicit None

TYPE(Loc), Intent(InOut) :: Location(:)             ! Array of TYPE: Loc
Character(Len=20), Intent(In):: Inventory    ! Name of the reanalysis inventory file
Real(dp):: NLAT, SLAT, WLON, ELON                   ! North/South Lat and E/W Lon boundaries
Real(dp):: Res                                                     ! Reanalysis Resolution of
global Lat/Lon matrix
Character(Len=30) :: A                                     ! Dummy Holder
Integer :: Counter                                                 ! Loop Counter

ierror1 = 0

! Open File and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'error Opening Inventory File in FillLoc Subroutine'

Do i = 1,4                                                            ! Move pointer over
the first 4 lines
       READ (20,*, IOSTAT = ierror1)
       If (ierror1 .NE. 0) EXIT
End Do

Read(20, *) A, A, NLAT, A, SLAT, A, Res             ! Read the 3rd, 5th, and 7th items in line 5
Read(20, *) A, WLON, A, ELON                        ! Read the 2nd and 4th items in line 6

Close (20)

! Manipulate the Lats/Lons into integers, loop through the values, and fill in the location array
Counter = 1

! This loop only works for Northern latitudes (Latitude is a postive number) and
! Westerly Longitudes (Longitude is given a a negative number)
Do i = Int(NLAT *10), Int(SLAT *10), -Int(Res * 10)
       Do j = Int(WLON *10), Int(ELON *10), Int(Res * 10)
              Location(Counter)%Lat = Real(i)/10
              Location(Counter)%Lon = Real(j)/10
              Counter = Counter + 1
       End Do
End Do

End Subroutine FillLocation
```

```
End Module LocationTools
```

TimeArray.f90

```
Module TimeTools

!*********************************************************************************************
!      Computes the locations of all reanalysis weather data within the spatial boundary
!
!      Date                  Programmers                           Description of Change
!      ====                  ==========                            ====================
!      25 OCT 05             MAJ Kevin Pace                 Original Code
!*********************************************************************************************

Use Kinds
Use Globals
Implicit None

Contains

Subroutine GetTimeSize (TimeSize, Inventory)
!*********************************************************************************************
! Extracts out # of unique Date-Time-Groups from the inventory file.  A typical inventory file
! lookslike:
!
!rec 1:0:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=1000 levels=(3,232) grid=255 1000 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 7 66  num bits 6  BDS_Ref 7  DecScale 0 BinScale 0
!
!rec 2:92:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=925 levels=(3,157) grid=255 925 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 674 730  num bits 6  BDS_Ref 674  DecScale 0 BinScale 0
!
!*********************************************************************************************

Use Kinds
Use Globals
Implicit None

Integer, Intent(InOut) :: TimeSize                   ! Size of Location Array
Character(Len=20), Intent(In):: Inventory    !Name of the reanalysis inventory file
Character(Len=3)   :: CheckRec                       ! First item of line.  Use to check if DTG
is on this line
Character(Len=200) :: A                              ! Dummy Holder
Integer :: TimeStamp1, TimeStamp2            ! TimeStamp in Inv file.  I made 2 for comparison
ability
Integer :: Arrow
ierror1 = 0

! Open File and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'error Opening Inventory File in GetTimeSize Subroutine'

TimeStamp1 = 1800000000              ! YYYYMMDDHH
TimeSize = 0
Rec = 0
Do             ! Read through each line of Inventory File
       Read (20,*, IOSTAT = ierror1) CheckRec ! Read first 3 characters of first object in line

       If (ierror1 .NE. 0) Then
              !Write(*,*) "GetTimeSize:  No first object in line.  EOR is found"
              Exit
       End If

       If (CheckRec .EQ. "rec") Then ! Is the line a record line (contains DTG)
              Backspace 20
              Rec = Rec + 1                  ! Sum up all records while we are counting
              Read (20,*, IOSTAT = ierror1) CheckRec, A, TimeStamp2 !First 3 objects of rec line
              If (ierror1 .NE. 0) Then
                     Write(*,*) "2. Error reading CheckRec in GetTimeSize Subroutine"
                     Exit
              End If
```

```
        If (TimeStamp2 .NE. TimeStamp1) Then ! Is this a new DTG?
                TimeSize = TimeSize +1   ! Add one to the array size
                TimeStamp1 = TimeStamp2 ! Make new DTG the old DTG for future comparisons
            End If
        End If

End Do

Close (20)

End Subroutine GetTimeSize


Subroutine FillTimeArray (DTG, Inventory)
!************************************************************************************************
!Fills the Location array with Lats/Lons in the order that the data file lists values.
!For reanalysis files, the first value listed is for the most NW location.  After that it moves
!across the Northern-most lat in an Eastward direction.  When it runs to the the most NE location
!it starts at the second most northern lat and the most western lon reading across in an Easterly
!direction.  It continues this 'typewriter' approach of assigning values when it reaches the most
!SE location.
!************************************************************************************************

Use Kinds
Use Globals
Implicit None

Integer, Intent(InOut) :: DTG(:)                    ! Array of Date-Time-Groups
Character(Len=20), Intent(In):: Inventory    ! Name of the reanalysis inventory file
Character(Len=3)   :: CheckRec                              ! First item of line.  Use to check
if DTG is on this line
Character(Len=200) :: A                                         ! Dummy Holder
Integer :: TimeStamp1, TimeStamp2                   ! TimeStamp in Inv file.  I made 2 for
comparison ability
Integer :: Counter

Counter = 1
ierror1 = 0

! Open File and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'error Opening Inventory File in GetTimeSize Subroutine'

TimeStamp1 = 1800000000              ! YYYYMMDDHH

Do                                                                                      ! Read
through each line of Inventory File
        Read (20,*, IOSTAT = ierror1) CheckRec ! Read first 3 characters of first object in line

        If (ierror1 .NE. 0) Then
            !Write(*,*) "GetTimeSize:  No first object in line.  EOR is found"
            Exit
        End If

        If (CheckRec .EQ. "rec") Then  ! Is the line a record line (contains DTG)
            Backspace 20
            Read (20,*, IOSTAT = ierror1) CheckRec, A, TimeStamp2 !First 3 objects of rec line
            If (ierror1 .NE. 0) Then
                Write(*,*) "2. Error reading CheckRec in GetTimeSize Subroutine"
                Exit
            End If
            If (TimeStamp2 .NE. TimeStamp1) Then ! Is this a new DTG?
                DTG(Counter) = TimeStamp2     ! Add one to the array size
                TimeStamp1 = TimeStamp2 ! Make new DTG the old DTG for future comparisons
                Counter = Counter + 1
            End If
        End If

End Do

Close (20)

End Subroutine FillTimeArray

End Module TimeTools
```

LevelArray.f90

```
Module LevelTools
```

```
!*********************************************************************************************
!      Computes the locations of all reanalysis weather data within the spatial boundary
!
!      Date                  Programmers                           Description of Change
!      ====                  ==========                            ====================
!      25 OCT 05             MAJ Kevin Pace                 Original Code
!*********************************************************************************************

Use Kinds
Use Globals
Implicit None

Contains

Subroutine GetLevelSize (LevelSize, Inventory)
!*********************************************************************************************
! Extracts out # of unique pressure levels from the inventory file.  A typical inventory file
! looks like (Pressure level is value after "kpds7=".  In this case, the first pressure level is
! 1000mb.  It is also towards the end of the first line as well:
!
!rec 1:0:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=1000 levels=(3,232) grid=255 1000 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 7 66  num bits 6  BDS_Ref 7  DecScale 0 BinScale 0
!
!rec 2:92:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=925 levels=(3,157) grid=255 925 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 674 730  num bits 6  BDS_Ref 674  DecScale 0 BinScale 0
!
!*********************************************************************************************

Use Kinds
Use Globals
Implicit None

Integer, Intent(InOut) :: LevelSize                  ! Size of Level Array
Character(Len=20), Intent(In):: Inventory    ! Name of the reanalysis inventory file
Character(Len=3)   :: CheckRec                               ! Used to check for a "rec" line in
data file
Character(Len=5)   :: A, B, C,VarNew,VarOld  ! A-C dummy; Var is variable for that record eg HGT
Integer            :: LvlSzTmp                               ! Pressure Level holders
Integer :: RecCounter, VarCounter                    ! Variables to ensure all records/variables
are read

VarCounter = 0 !Result should be #DTGs * (# Variables +1) -> Hgt, UGRD, VGRD, TMP, RH, + Pressure
RecCounter = 0 !Result should be # Records in file
ierror1   = 0
LevelSize = 0
LvlSzTmp  = 1
VarOld = "KEVIN"       !Initialize VarOld to something that will never appear in Inventory File

! Open File and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'error Opening Inventory File in GetLevelSize Subroutine'


Do                     ! Read through each line of Inventory File
       Read (20,*, IOSTAT = ierror1) CheckRec ! Read first 3 characters of first object in line
       If (ierror1 .NE. 0) Then
              !Write(*,*) "GetLevelSize:  No first object in line.  EOR is found"
              Exit
       End If

       If (CheckRec .EQ. "rec") Then         ! Is line a record line (contains pressure level)
              Backspace 20                   ! Back up to the "rec" line that we just read
              RecCounter = RecCounter +1
              Read (20,*, IOSTAT = ierror1) A,B,C,VarNew   ! Get first 4 objects of rec line
              If (ierror1 .NE. 0) Then
                     Write(*,*) "2. Error reading CheckRec in GetLevelSize Subroutine"
                     Exit
              End If
```

```
        If (VarNew .EQ. VarOld) Then   ! Is this a new variable?
            LvlSzTmp = LvlSzTmp + 1 ! Sum up levels for this particular variable
        Else
            If(LvlSzTmp .GT. LevelSize) LevelSize = LvlSzTmp
            LvlSzTmp = 1
            VarOld = VarNew
            VarCounter = VarCounter + 1
        End If
    End If
End Do

Close (20)

Write(*,*) "There are ", LevelSize, " pressure levels covered by this file"
Write(*,*) "The GetLevelSize Sub read", RecCounter, " records and ", VarCounter,"variables"

End Subroutine GetLevelSize


Subroutine FillLevelArray (Level, Inventory, LevelSize)
!*****************************************************************************************
!Fills the Location array with Lats/Lons in the order that the data file lists values.
!For reanalysis files, the first value listed is for the most NW location.  After that it moves
!across the Northern-most lat in an Eastward direction.  When it runs to the the most NE location
!it starts at the second most northern lat and the most western lon reading across in an Easterly
!direction.  It continues this 'typewriter' approach of assigning values when it reaches the most
!SE location.
!*****************************************************************************************

Use Kinds
Use Globals
Implicit None

Integer, Intent(InOut) :: Level(:)          ! Array of Pressure Levels [mb]
Character(Len=20), Intent(In):: Inventory   ! Name of the reanalysis inventory file
Integer, Intent(In):: LevelSize             ! This is the size of the Level array
Character(Len=3)   :: CheckRec              ! Used to check for a "rec" line in data file
Integer                 :: PrField          ! Holds Pressure Level Object e.g. 995
Character(Len=5)   :: A,B,C,D,E,F,G,H,L ! dummy variables - used as placeholders and debugging
Character(Len=5)   :: VarNew,VarOld         ! Var is variable for that record (eg HGT)
Integer            :: Counter1              ! # Times we have read in a value to Level Array
Integer                 :: LvlSzTmp         ! # of levels we have read for the current variable


ierror1  = 0
VarOld   = "KEVIN"     !Initialize VarOld to something that will never appear in Inventory File
LvlSzTmp = 1

! Open File and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'Error Opening Inventory File in FillLevelArray Subroutine'

Do       ! Read records until you find a variable with values at all pressure levels
    Read (20,*, IOSTAT = ierror1) CheckRec ! Read first 3 characters of first object in line
    If (ierror1 .NE. 0) Then
        !Write(*,*) "GetLevelSize:  No first object in line.  EOR is found"
        Exit
    End If

    If (CheckRec .EQ. "rec") Then         !Is line a record line (contains pressure level)

        Backspace 20                  !Back up to the "rec" line that we just read

        Read (20,*, IOSTAT = ierror1) A,B,C,VarNew   ! Get first 4 objects of rec line
        If (ierror1 .NE. 0) Then
            Write(*,*) "2. Error reading CheckRec in GetLevelSize Subroutine"
            Exit
        End If

        If (VarNew .EQ. VarOld) Then          !Same variable as the last record we read?
            LvlSzTmp = LvlSzTmp + 1       !Sum up levels for this particular variable

            If(LvlSzTmp .EQ. LevelSize) Then !Levels for this variable = level size?
                Do i = 1, (LevelSize * 8) !If so, backup to rec where var starts
                    BackSpace 20   ! and get out of this loop
                End Do
                Exit
            End If
        Else                          ! If not new variable, we will
            LvlSzTmp = 1          ! Start the level counter over and
            VarOld = VarNew       ! Make comparison variable equal to new variable
```

```
            End If
        End If
End Do

Counter1=0

Do              ! Read through records until we find a variable with "LevelSize" contiguous
                ! records. The pointer should start at the first record of the first variable
                ! that is defined at all levels.  For Pressure levels, its probable HGT

        Read (20,*, IOSTAT = ierror1) CheckRec ! Read first 3 characters of first object in line
        If (ierror1 .NE. 0) Then
                Write(*,*) "FillLevelArray:  No first object in line.  EOR is found"
                Exit
        End If

        If (CheckRec .EQ. "rec") Then  ! Is line a record line (contains pressure level)

                Backspace 20           ! Back up to the "rec" line that we just read

                ! Get first 10 objects of rec line (#4 is Variable, #10 is pressure in millibars
                Read (20,*, IOSTAT = ierror1) CheckRec,B,C,VarNew,D,E,F,G,H,L,PrField
                If (ierror1 .NE. 0) Then
                        Write(*,*) "2. Error reading CheckRec in FillLevelArray Subroutine"
                        Exit
                End If

                Counter1 = Counter1 + 1       ! Sum up levels for this particular variable
                Level(Counter1) = PrField

                If (Counter1 .EQ. LevelSize) Exit
        End If
End Do

Close (20)
Write(*,*) "FillLevelArray levels were determined by ", VarOld
Write(*,*) "This variable sequence ended on record ", B

End Subroutine FillLevelArray

End Module LevelTools
```

WxArray.f90

```
Module WxTools

!*********************************************************************************************
!       Fills WXPT array with values.  Values come from the data file as opposed to the inventory
!   file.
!
!       Date                  Programmers                          Description of Change
!       ====                  ==========                           =====================
!       27 OCT 05             MAJ Kevin Pace               Original Code
!*********************************************************************************************

Use Kinds
Use Globals
Implicit None

Contains

Subroutine FillWxPTArray (WXPT, Level, Location, DTG, Inventory, DataFile)
!*********************************************************************************************
! This Subroutine reads the WxPT data from the datafile using the Inventory file for an
! explanation of what each block of numbers mean.  For example, the first few lines of a
! typical datafile look like:
!  4 3
!  7
!  6
!  12
!  10
!  3
!  3
!  27
!  6
!  9
!  8
!  8
!  6
```

```
!  4 3
! This means that for the 4x3 matrix of locations, these are the values of what record 1 in
! in the Inventory file describe.  The first few lines of a typical reanalysis inventory file
! look like:
!
!rec 1:0:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=1000 levels=(3,232) grid=255 1000 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 7 66  num bits 6  BDS_Ref 7  DecScale 0 BinScale 0
!
!rec 2:92:date 1952060100 HGT kpds5=7 kpds6=100 kpds7=925 levels=(3,157) grid=255 925 mb anl:
!  HGT=Geopotential height [gpm]
!  timerange 10 P1 0 P2 0 TimeU 1  nx 4 ny 3 GDS grid 0 num_in_ave 0 missing 0
!  center 7 subcenter 1 process 80 Table 2 scan: WE:NS winds(N/S)
!  latlon: lat  40.000000 to 35.000000 by 2.500000  nxny 12
!          long -120.000000 to -112.500000 by 2.500000, (4 x 3) scan 0 mode 128 bdsgrid 1
!  min/max data 674 730  num bits 6  BDS_Ref 674  DecScale 0 BinScale 0
!
! So in the above example, the 12 values under "4 3" are the heights of the 1000mb pressure
! level.  The values are ordered in a sequence like a typewriter:  NW to SE lat/lon locations.
! So at location 40 lat/-120 lon the height of the 1000mb level is 7m and the height of the
! 1000mb pressure level at 35 lat/-112.5 lon is 6m.
!*********************************************************************************************

Use Kinds
Use Globals
Implicit None

Type(WxPoint),Intent(InOut):: WxPT(:,:,:)! 3D array of weather data points
Integer,Intent(In)         :: Level(:)        ! Pressure levels for which Wx data is avail [mb]
Type(Loc),Intent(In)       :: Location(:)! Locations for which Wx data is avail [Lon, Lat]
Integer,Intent(In)        :: DTG(:)    ! Date-Time-Groups where Wx data is avail [YYYYMMDDHH]
Character(Len=20), Intent(In):: Inventory, DataFile  !Name of the reanalysis files
Character(Len=5):: CheckRec,Var              ! Key Fields from rec line of Inv File
Integer                :: Lvl                ! Key Fields from rec line of Inv File
Real(dp)               :: Time               ! Key Fields from rec line of Inv File
Character(Len=5):: B,E,F,G,H,L,M             ! Dummy Variables between fields and for debugging
Real(dp)               :: Temp(1:LocSize)    ! Temporary Array holding sets of data from datafile
Character(Len=10)::Flag, CheckFlag           ! FLAG separates data groups in datafile
Integer                :: DI, LI             ! DTG index and Level Index for array searching

ierror1 = 0
ierror2 = 0

! Open Files and check for errors on OPEN
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'Error Opening Inventory File in FillWxPTArray Subroutine'
OPEN (UNIT = 30, FILE=DataFile,  STATUS='OLD', ACTION='READ', IOSTAT=ierror2)
If (ierror2 .NE. 0) Write(*,*) 'Error Opening Data File in FillWxPTArray Subroutine'

Do      i = 1,(rec*7) !Read every record in the Inv and Data File and extract key fields of data
        !********Get key fields from Inv File****************************************************
        Read (20,*, IOSTAT = ierror1) CheckRec ! Read first 3 characters of first object in line
        If (ierror1 .NE. 0) Then
                Write(*,*) "FillWxPTArray:  No first object in line in Inv File.  EOR is found"
                Exit
        End If

        If (CheckRec .EQ. "rec") Then         ! Is line a record line (contains pressure level)
                Backspace 20                  ! Back up to the "rec" line that we just read

                ! Get Time, Variable, and PressureLevel
                Read (20,*, IOSTAT = ierror1) CheckRec,B,Time,Var,E,F,G,H,L,M,Lvl
        !*****************************************************************************************

                !*****Find index of "Time" in DTG array and index of "Level" in Level Array*******
                Do j = 1, TimeSize
                        If(DTG(j) .EQ. Time) Then
                                DI = j
                                Exit
                        End If
                End Do

                Do j = 1, LevelSize
                        If(Level(j) .EQ. Lvl) Then
                                LI = j
                                Exit
                        End If
```

```
        End Do

    !*************************************************************************************

        !**Read the group of data from datafile that corresponds to the record in Inv File
        Do
        Read (30,'(a)', IOSTAT = ierror2) CheckFlag  !Read a line from DataFile
        If (ierror2 .NE. 0) Then
           Write(*,*) "FillWxPTArray: No first object in line of datafile. EOR is found"
           Exit
        End If

        If (i .EQ. 1) Flag = CheckFlag !First item of every data file is the flag. Do once

        If (CheckFlag .EQ. Flag) Then         ! Is line a value separator in the data file
            Do j = 1, LocSize
                Read (30,*, IOSTAT = ierror1) Temp(j) !Store value in Temp Array
            End Do
            Exit
        End If
        End Do

    !*************************************************************************************

    !********Assign the datafile data set into the appropriate place in WxPT array
        SelectCase (Trim(Var))
            Case("HGT")
                        WxPT(LI,:,DI)%HGT = Temp(:)
            Case("UGRD")
                        WxPT(LI,:,DI)%U   = Temp(:)
            Case("VGRD")
                        WxPT(LI,:,DI)%V   = Temp(:)
            Case("TMP")
                        WxPT(LI,:,DI)%T   = Temp(:)
            Case("RH")
                        WxPT(LI,:,DI)%RH  = Temp(:)
        End Select

    !*************************************************************************************
    End If

End Do
    !*************************************************************************************
Close (20)
Close (30)

! Fill up the WxPT%WndSpd and WxPT%WndDir in the WxPT Array
Call UVConverter (WxPT)

End Subroutine FillWxPTArray



Subroutine UVConverter (WxPT)
!*********************************************************************************************
! Converts the U- and V- wind speeds into a windspeed and direction.  Basically, I am
! taking WXPT%U and WXPT%V and calculating WXPT%WndSpd and WXPT%WndDir
!*********************************************************************************************
Use Kinds
Use Globals
Implicit None

Type(WxPoint),Intent(InOut):: WxPT(:,:,:)! 3D array of weather data points
Real(dp):: CartDeg     !Cartesian Degree described by U,V components of the wind
Integer :: Quadrant    !Quadrant in which the Cartesian Degree resides (I = Upper Right
                               !II = Upper Left, III = Lower Left, and IV = Lower Right)
Real(dp):: WndDir


!Compute WindSpeed (The Easy Part).  This can be done directly on the entire matrix
WxPT%WndSpd = SQRT(WxPT%V**2 + WxPT%U**2)

!Compute WindDirection.  In Cartesian Coordinates positive angles are measured from the
!positive X-axis (0 degrees) in a CounterClockWise (CCW) direction.  In meteorology, the
!angles are positive from the positive Y-axis (called V (Northern Direction)) in a ClockWise
!Direction.
Do i = 1,TimeSize
    Do j = 1, LevelSize
        Do k = 1, LocSize

            !First we get the angles in normal Cartesian values
```

```
        !This is Degrees from -180 to 180 (0 is East(U), and angles are positive going CCW)
        !ATAN2D takes a Y,X (or V,U) pair as arguements
                        CartDeg =  ATAN2D(WxPT(j,k,i)%V,WxPT(j,k,i)%U)

                        !Transform Cartesian angle (-180 to 180) to Cartesian angle (0 to 360)
                        If(CartDeg .GE. 0._dp .AND. CartDeg .LE. 180._dp) Then
                                CartDeg = CartDeg
                        Else If(CartDeg .LT.  0._dp .AND. CartDeg .GT. -180._dp) Then
                                CartDeg = CartDeg + 360._dp
                        Else If(CartDeg .EQ.  -180._dp) Then
                                CartDeg = 180._dp
                        Else
                                Write(*,*) "UV Conv Cart: Angle input not between -180 and 180"
                        End If

                        !Find the quadrant of this angle
                        If(CartDeg .GE. 0._dp .AND. CartDeg .LE. 90._dp) Then
                                Quadrant = 1
                        Else If(CartDeg .GT.  90._dp .AND. CartDeg .LE. 180._dp) Then
                                Quadrant = 2
                        Else If(CartDeg .GT. 180._dp .AND. CartDeg .LE. 270._dp) Then
                                Quadrant = 3
                        Else If(CartDeg .GT. 270._dp .AND. CartDeg .LT. 360._dp) Then
                                Quadrant = 4
                        Else If(CartDeg .EQ.  360._dp) Then
                                CartDeg = 0.0_dp
                                Quadrant = 1
                        Else
                                Write(*,*) "UV Conv Quadrant: Angle input not between 0 and 360"
                        End If

                        ! Turn Cartesian Angle (0-359 going CCW from +X axis)
                        ! into Azimuthal Angle (0-360 going CW from +Y axis)

                        SelectCase (Quadrant)
                                Case(1)
                                        WxPT(j,k,i)%WndDir =  90._dp - CartDeg
                                Case(2)
                                        WxPT(j,k,i)%WndDir = 450._dp - CartDeg
                                Case(3)
                                        WxPT(j,k,i)%WndDir = 450._dp - CartDeg
                                Case(4)
                                        WxPT(j,k,i)%WndDir = 450._dp - CartDeg
                                Case Default
                                        Write(*,*) "UV Conv WndDir: Angle not in Quadrant I-IV"
                                End Select

                        ! Now convert this angle to where the wind is coming from NOT GOING TO!
        If (WxPT(j,k,i)%WndDir .EQ. 0._dp ) Then
                WxPT(j,k,i)%WndDir = 180._dp
        Else If (WxPT(j,k,i)%WndDir .GT. 0._dp .AND. WxPT(j,k,i)%WndDir .LT. 180._dp ) Then
                WxPT(j,k,i)%WndDir = WxPT(j,k,i)%WndDir + 180._dp
        Else If (WxPT(j,k,i)%WndDir .EQ. 180) Then
                WxPT(j,k,i)%WndDir = 0._dp
        Else If (WxPT(j,k,i)%WndDir .GT. 180._dp .AND. WxPT(j,k,i)%WndDir .LE. 360._dp) Then
                WxPT(j,k,i)%WndDir = WxPT(j,k,i)%WndDir - 180._dp
        Else
                Write(*,*) "UV Converter:  Angle Reversal not working.  Input angle not 0-360!"
        End If
                End Do
        End Do
End Do

End Subroutine UVConverter

End Module WxTools
```

HPAC_PRF_WRITER.f90

```
Module PrfWriter

!**********************************************************************************************
!       Writes a textfile with a .prf extension.  This file contains weather data in a format in
!   which HPAC can ingest it.
!
!       Date                  Programmers                                  Description of Change
!       ====                  ==========                                   =====================
!       29 OCT 05             MAJ Kevin Pace                                       Original Code
!**********************************************************************************************
```

```
Use Kinds
Use Globals
Implicit None

Contains

Subroutine WritePRF (WxPT, DTG, Location, Level, Inventory)
!*********************************************************************************************
!       This Subroutine takes the newly-filled WxPT array and writes the data into a format
!       in which FORTRAN can understand.  This format is explained in the HPAC 4.03 user manual
!   starting on page 573.  The basic gist is copied here for a quick explanation:
!
!       Profile File
!       A sample PRF file is shown below.
!
!       # CREATOR: WXEDITOR
!       # DATE: 2001-04-17 20:58:42 GMT
!       # SOURCE: obs
!       # REFERENCE: agl
!       # TYPE: OBSERVATION
!       # ANALYSIS: 2001 04 17 12.00
!       # START: 2001 04 17 12.00
!       # END: 001 04 17 15.00
!       # TIMEREFERENCE: UTC
!       # MODE: profile all
!       PROFILE
!       8 6
!       ID YYMMDD HOUR LAT LON ELEV ZI HFLUX
!       HOURS N E M M W/M2
!       Z WDIR WSPD P T H
!       M DEG M/S MB C %
!       HPAC 4.04 User's Manual
!       574
!       -9999
!       ID: 722650 010417 12.00 31.95 -102.22 872 112 -28.68
!       2 360 5.1 960 2.6 97
!       680 20 19.0 925 3.8 100
!       1369 45 14.9 850 4.2 100
!       2933 55 12.9 700 -2.9 94
!
!Header lines begin with the # character in the first column. All header lines are at the
!beginning of the PRF file. As shown above, the header lines describe the data type
!(Observation, Forecast, or Analysis), the time reference (i.e., UTC or LOCAL), which
!application wrote the file and when it was written. For Forecast files, an Analysis header
!line will appear defining the date and time of the model analysis.
!The keyword entry PROFILE indicates that this is an upper air observations file.
!The first number 6 indicates there are six Fixed Data columns in the ID line of the PRF file.
!The Fixed Data columns contain data that refer to the observing station. The second number 6
!indicates there are six Profile Data columns. Profile Data columns contain the multi-level,
!upper air observations.  The first two lines list the Fixed Data variable names and the units
!for each fixed data variable  respectively. The Fixed Data are given once for each report. A
!summary of the Fixed Data variable names and units typically used in the PRF files is given in
!the table below.
!
!Fixed Data Variable Description     Fixed Data Variable Name      Fixed Data Variable Units
!       Station ID                   ID                            None
!       Year-Month-Day               YYMMDD                        None
!       Hour                         HOUR                          HOURS
!       Latitude                     LAT                           N
!       Longitude                    LON                           E
!       Station Elevation            ELEV                          M
!
!The last two lines list the Profile Data variable names and the units for each Profile Data
!variable respectively. The Profile Data are given for each level in the report. A summary of the
!Profile Data variable names and units typically used in the PRF files is given in the table
!below.
!
!       Profile Data Variable        Profile Data                  Profile Name
!       Description                  Variable Name                 Variable Units
!       Altitude                     Z                             M
!       Wind Direction               WDIR (or DIR)                 DEG
!       Wind Speed                   WSPD (or SPEED or SPD)        M/S
!       Pressure                     P                             MB
!       Temperature                  T                             C
!       Humidity                     H (or HUMID or Q)             %
!
!
!       The number -9999 is the indicator used for missing data.
!       The output file contains the data values in column order. All observations for a
!   particular station, date, and time are grouped together. Within a group, the observations
!   are listed in order of ascending height. When observations are available for multiple
```

```
!stations or multiple times the output file will contain multiple sections that are similar
!to the above example. Each of these sections will begin with a unique ID: line.
!
!*************************************************************************************

Use Kinds
Use Globals
USE DFPORT
Implicit None

Type(WxPoint),Intent(In)    :: WxPT(:,:,:)    ! 3D array of weather data points
Integer,  Intent(In)        :: Level(:)       ! Pressure levels for which Wx data is avail [mb]
Type(Loc),Intent(In)        :: Location(:)    ! Locations for which Wx data is avail [Lon, Lat]
Integer,  Intent(In)        :: DTG(:)  ! Date-Time-Groups where Wx data is avail [YYYYMMDDHH]
Character(Len=20), Intent(In):: Inventory
Character(Len=20)            :: OutputFile    ! Name of prf file that will be created
Character(Len=3)             :: Initials      ! Name of person creating file
Integer :: Year, Month, Day, Elev             ! DTG index and Level Index for array searching
Character(Len=24):: CurrentTime
Integer:: TimeArray(8)                        ! Arrat containing time information
Character(Len =10)::Analysis, StartTime, EndTime
Integer :: ID1, ID2    , LonGrid, LatGrid, Arrow     ! Used for finding LonGrid and LatGrid
Character(Len=200):: Line6                     ! Used for finding LonGrid and LatGrid
Character(Len = 6):: IDNumber                 ! ID Number
Character(Len = 2)::  F3, L3                  ! First 3 numbers, Last 3 Numbers of IDNumber


ierror1 = 0

Write(*,*) 'Enter name of HPAC .prf file to be created (exampl: george.prf): '
Read(*,*)  OutputFile
!OutputFile = "Output.txt"

Write(*,*) 'Enter your initials (limited to 3 letters): '
Read(*,*)  Initials
!Initials = "kpd"     !Initials of Kevin David Pace


!********Get LatGrid and LonGrid for writing to "ID:" field ****************************
OPEN (UNIT = 20, FILE=Inventory, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'error Opening Inventory File in GetLocSize Subroutine'

Do i = 1,5                    ! Move pointer over the first 5 lines
       READ (20,*, IOSTAT = ierror1)
       If (ierror1 .NE. 0) EXIT
End Do

Read (20,'(a)', IOSTAT = ierror1) Line6             ! Read the 6th Line of the Inventory file
arrow = index(Line6,"(") + 1                        ! Find the ( before the Number of Lons
Read (Line6(arrow:),*) LonGrid                      ! Read the number of Lons in the grid
arrow = index(Line6,"x") + 1                        ! Find the x before the Number of Lats
Read (Line6(arrow:),*) LatGrid                      ! Read the number of Lats in the grid

Close (20)
!**********************************************************************************

OPEN (UNIT = 40, FILE=OutputFile, STATUS='NEW', ACTION='WRITE', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'WritePRF Subroutine: Error Creating PRF file'

Write(40, 5000) Initials
5000 Format ("# CREATOR:", T19, A3)

CurrentTime = FDate()
Write(40,5010) CurrentTime
5010 Format ("# DATE:", T19, A24, 1x, "Local")

Write(40,5020) "GRIB"
5020 Format ("# SOURCE:", T19, A4)

Write(40,5030) "no"
5030 Format ("# EDITED:", T19, A2)

Write(40,5040) "msl"
5040 Format ("# REFERENCE:", T19, A3)

Write(40,5050) "forecast"
5050 Format ("# TYPE:", T19, A8)

Write(Analysis, '(i10)') (DTG(1))
Write(40,5060) Analysis(1:4), Analysis(5:6), Analysis(7:8),Analysis(9:10)
5060 Format ("# ANALYSIS:", T19, A4, 1x, A2, 1x, A2, 1x, A2, ".00")
```

```
Write(StartTime, '(i10)') (DTG(1))
Write(40,5070) StartTime(1:4), StartTime(5:6), StartTime(7:8),StartTime(9:10)
5070 Format ("# START:", T19, A4, 1x, A2, 1x, A2, 1x, A2, ".00")

Write(EndTime, '(i10)') (DTG(TimeSize))
Write(40,5080) EndTime(1:4), EndTime(5:6), EndTime(7:8),EndTime(9:10)
5080 Format ("# END:", T19, A4, 1x, A2, 1x, A2, 1x, A2, ".00")

Write(40,5090) "UTC"
5090 Format ("# TIMEREFERENCE:", T19, A3)

Write(40,5100) "Profile All"
5100 Format ("# MODE:", T19, A11)

Write(40,5110) "PROFILE"
5110 Format (A7)

Write(40, 5120) 6, 6
5120 Format (I1, 1x, I1)

Write(40, 5130) "ID      ", "YYYYMMDD  ", "HOUR    ", "LAT     ", "LON     ", "ELEV    "
5130 Format (A8, A8, A8, A8, A8, A8)

Write(40, 5140) "HOURS   ", "N       ", "E       ", "M       "
5140 Format (T17, A8, A8, A8, A8)

Write(40, 5150) "Z       ", "WDIR    ", "WSPD    ", "P       ", "T       ", "HUMID   "
5150 Format (A8, A8, A8, A8, A8, A8)

Write(40, 5160) "M       ", "DEG     ", "M/S     ", "MB      ", "K       ", "%       "
5160 Format (A8, A8, A8, A8, A8, A8)

Write(40, 5170) -9999
5170 Format (I5)

!Step Through Time Blocks
Do i = 1, TimeSize

Write(StartTime, '(i10)') (DTG(i))

ID1 = 0
ID2 = 1

        !Step Through Levels
        Do j = 1, LocSize
                Call Elevation(Location(j)%Lat,Location(j)%Lon,Elev)

        !*********This block of code constructs the ID # for the .prf file*********************
                IDNumber = "000000"
                ID1 = ID1 +1            !Gets the ID numbering sequence
                If(ID1 .EQ. LonGrid + 1) Then
                        ID1 = 1
                        ID2 = ID2 +1
                End If

                Write(F3, '(I2)') ID1
                Write(L3, '(I2)') ID2

                If (ID1 .GE. 10) Then
                        IDNumber(2:3) = F3(1:2)
                Else
                        IDNumber(3:3) = F3(2:2)
                End If

                If (ID2 .GE. 10) Then
                        IDNumber(5:6) = L3(1:2)
                Else
                        IDNumber(6:6) = L3(2:2)
                End If

        !*****************************************************************************************

                Write(40, 5180) "ID: ",IDNumber, StartTime(1:8),StartTime(9:10),Location(j)%Lat, &
                & Location(j)%Lon, Elev
                5180 Format (A4, T5, A6, T14, A8, T23, A2, ".00", T30, F7.4, T38, F9.4,T50, I5)

                Do k = 1, LevelSize
                !Write(40, *) k, j, i
                !Write(40,*) WxPT(k,j,i)%HGT
                        Write(40, 5190) Int(WxPT(k,j,i)%HGT), Int(WxPT(k,j,i)%WndDir), &
```

```
                      & WxPT(k,j,i)%WndSpd, Int(Level(k)),WxPT(k,j,i)%T, Int(WxPT(k,j,i)%RH)
                      5190 Format (3x, I5, T10, I3, T19, F5.1, T30, I4, T43, F5.1, T52, I5)
               End Do
        End Do
End Do

Close (40)

End Subroutine WritePRF


Subroutine Elevation (Lat, Lon, Elev)
!**********************************************************************************************
!This Subroutine takes a Latitude and Longitude and returns the surface elevation.  It does
!this by looking up the elevation from a data file (Surface Elevation.dat) which was created
!by saving an Excel File as a Unicode text file.  The data for this file comes from Google
!Earth.  This data is limited to 36 locations.  The locations range from (42.5,-122.5) to
!(30,-110) in a 2.5 degree resolution,
!
!       Surface Elevation.dat Contents:
!
!       Latitude       Longitude      Elevation (ft) Elevation (m)
!       42.5           -122.5         2821                 860.0609756
!       42.5           -120           6000                 1829.268293
!       42.5           -117.5         4913                 1497.865854
!       42.5           -115           4271                 1302.134146
!       42.5           -112.5         6861                 2091.768293
!       42.5           -110           6979                 2127.743902
!       40                    -122.5         792                         241.4634146
!       40                    -120           6233                 1900.304878
!       40                    -117.5         5541                 1689.329268
!       40                    -115           6293                 1918.597561
!       40                    -112.5         6857                 2090.54878
!       40                    -110           5336                 1626.829268
!       37.5           -122.5         319                         97.25609756
!       37.5           -120           2706                 825
!       37.5           -117.5         6878                 2096.95122
!       37.5           -115           5456                 1663.414634
!       37.5           -112.5         7334                 2235.97561
!       37.5           -110           6166                 1879.878049
!       35                    -122.5         0                           0
!       35                    -120           4436                 1352.439024
!       35                    -117.5         2423                 738.7195122
!       35                    -115           2815                 858.2317073
!       35                    -112.5         5386                 1642.073171
!       35                    -110           5393                 1644.207317
!       32.5           -122.5         0                           0
!       32.5           -120           0                           0
!       32.5           -117.5         0                           0
!       32.5           -115           85                          25.91463415
!       32.5           -112.5         2432                 741.4634146
!       32.5           -110           4386                 1337.195122
!       30                    -122.5         0                           0
!       30                    -120           0                           0
!       30                    -117.5         0                           0
!       30                    -115           1895                 577.7439024
!       30                    -112.5         1403                 427.7439024
!       30                    -110           3622                 1104.268293
!
!**********************************************************************************************

Use Kinds
Use Globals
Implicit None

Real(dp),Intent(In)     :: Lat        ! Degrees of North Latitude
Real(dp),Intent(In)     :: Lon        ! Degrees of Eastern Longitude
Integer,Intent(InOut):: Elev          ! Surface Elevation in meters
Real(dp)                       :: TempLat, TempLon ! Lats and Lons read from data file
Character(Len=22)       :: SfcFile    ! Name of .dat file containing surface elevation data
Integer                        :: Dummy1     ! Elevation (in Feet) from .dat file
Real(dp)                       :: TempElev   ! Elevation (in meters) from .dat file

SfcFile = "Surface Elevation.dat"
ierror1 = 0

OPEN (UNIT = 50, FILE=SfcFile, STATUS='Old', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'Elevation Subroutine: Error opening surface elevation data file'

Read (50,*)            ! Read over 1st line in data file (its header data)
```

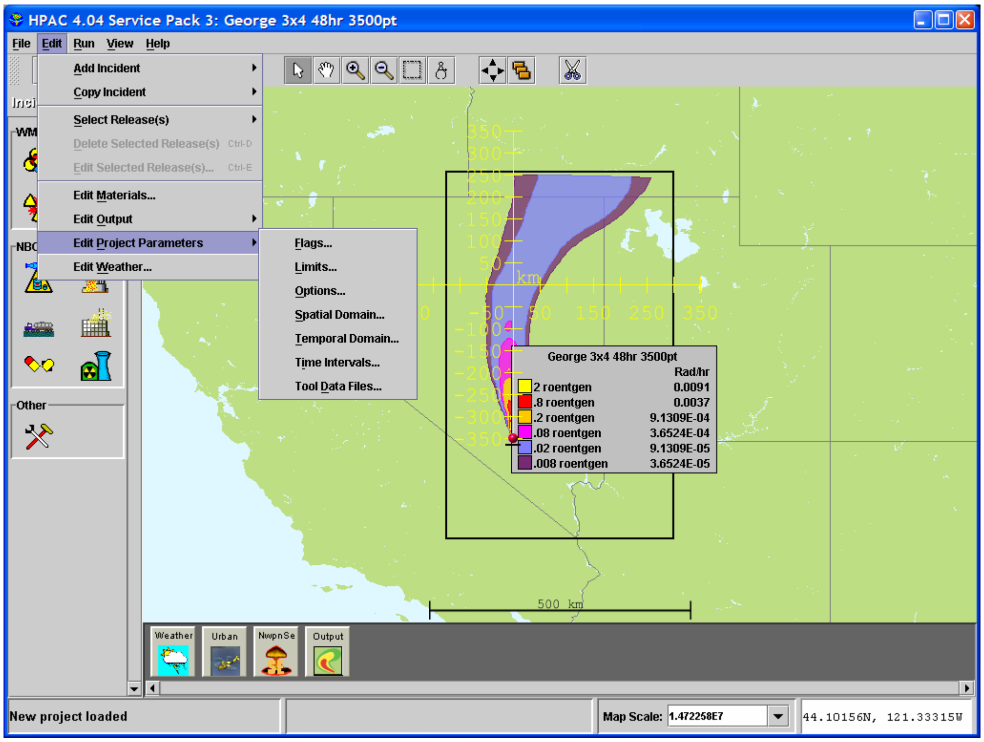
```
Do      ! Sequentially read through the records

        Read (50,*, IOSTAT = ierror1) TempLat, TempLon, Dummy1, TempElev    !Read .dat file record
        If (ierror1 .NE. 0) Then
                Write(*,*) 'Elevation Subroutine: Error reading a record in Elevation data file'
                Exit
        End If

        If (TempLat .EQ. Lat .AND. TempLon .EQ. Lon) Then    ! Is this the right location?
                Elev = TempElev
        ! Get Elevation
                Exit
        ! Exit and pass Elev to WritePRF Sub
        End If
End Do

Close (50)

End Subroutine Elevation


End Module PrfWriter
```

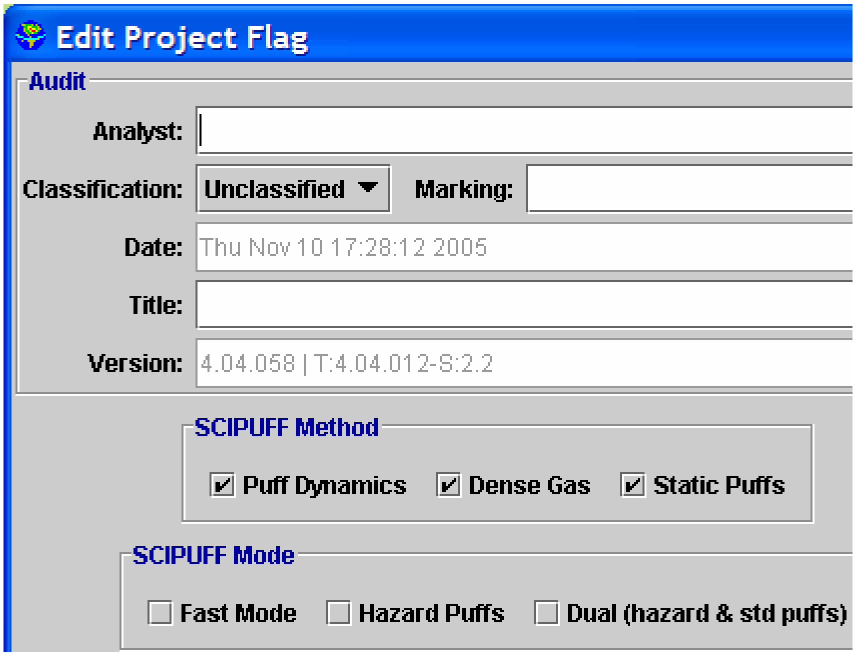
Surface Elevation.dat

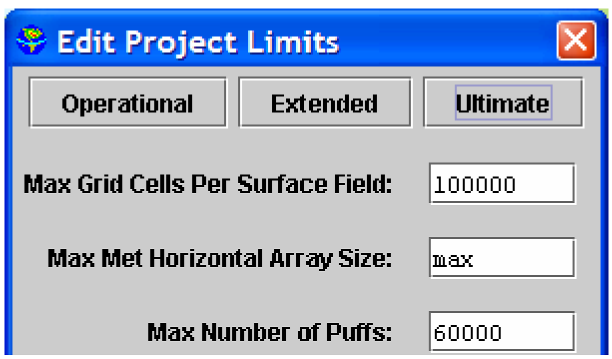
Latitude	Longitude	Elevation (ft)	Elevation (m)
42.5	-122.5	2821	860.0609756
42.5	-120	6000	1829.268293
42.5	-117.5	4913	1497.865854
42.5	-115	4271	1302.134146
42.5	-112.5	6861	2091.768293
42.5	-110	6979	2127.743902
40	-122.5	792	241.4634146
40	-120	6233	1900.304878
40	-117.5	5541	1689.329268
40	-115	6293	1918.597561
40	-112.5	6857	2090.54878
40	-110	5336	1626.829268
37.5	-122.5	319	97.25609756
37.5	-120	2706	825
37.5	-117.5	6878	2096.95122
37.5	-115	5456	1663.414634
37.5	-112.5	7334	2235.97561
37.5	-110	6166	1879.878049
35	-122.5	0	0
35	-120	4436	1352.439024
35	-117.5	2423	738.7195122
35	-115	2815	858.2317073
35	-112.5	5386	1642.073171
35	-110	5393	1644.207317
32.5	-122.5	0	0
32.5	-120	0	0
32.5	-117.5	0	0
32.5	-115	85	25.91463415
32.5	-112.5	2432	741.4634146
32.5	-110	4386	1337.195122
30	-122.5	0	0
30	-120	0	0
30	-117.5	0	0
30	-115	1895	577.7439024
30	-112.5	1403	427.7439024
30	-110	3622	1104.268293

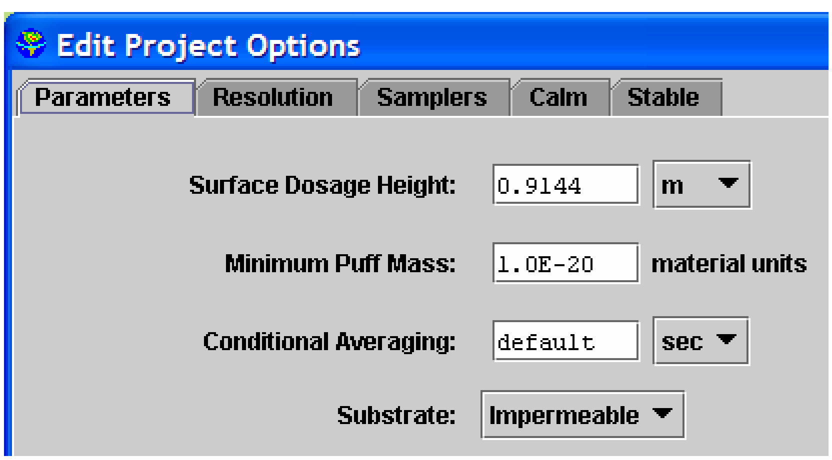
Appendix D: HPAC Project Parameters

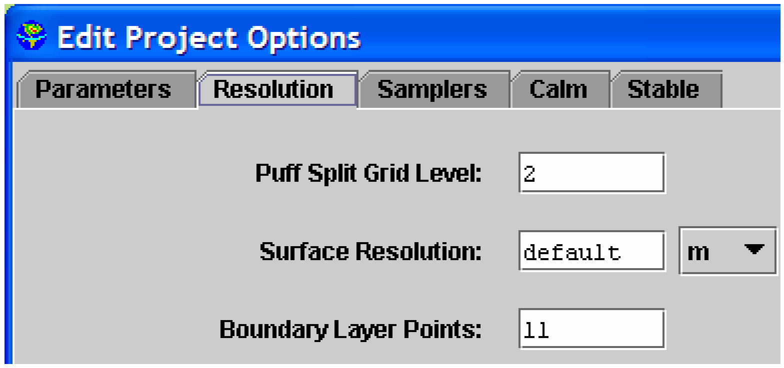
This appendix documents the settings used when running HPAC simulations. The settings shown in this appendix focus on the settings in the “Edit Project Parameters” area of HPAC.

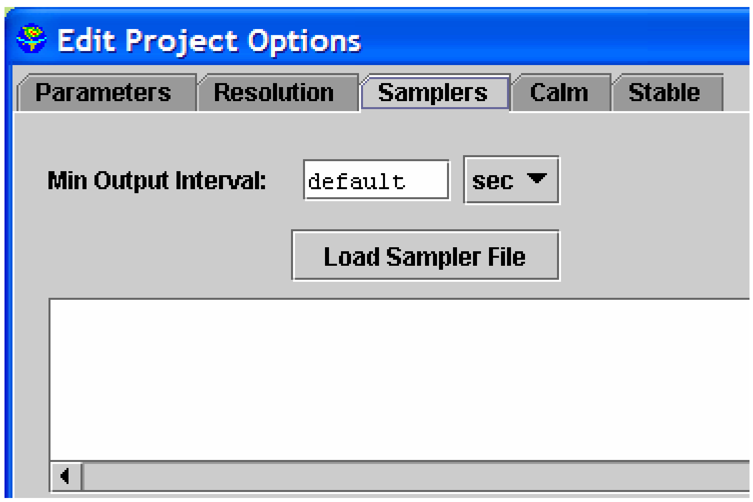


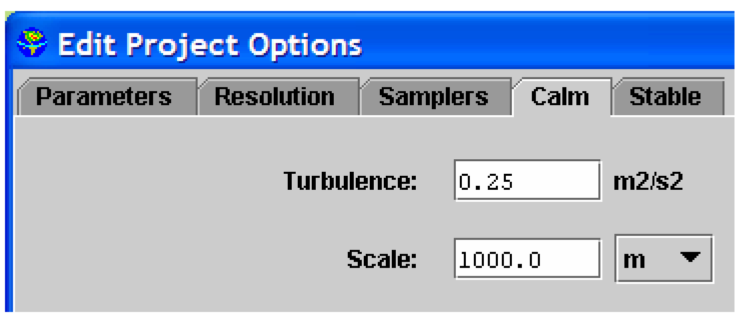

Edit Project Flag
Audit
Analyst:
Classification: Unclassified
Marking:
Date: Thu Nov 10 17:28:12 2005
Title:
Version: 4.04.058 | T:4.04.012-S:2.2
SCIPUFF Method
Puff Dynamics
Dense Gas
Static Puffs
SCIPUFF Mode
Fast Mode
Hazard Puffs
Dual (hazard & std puffs)

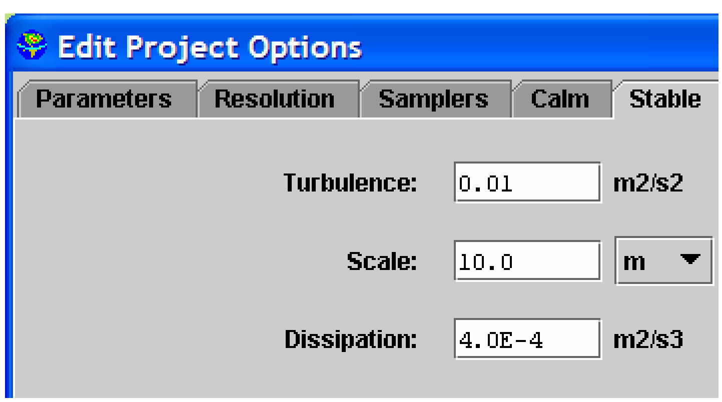

Edit Project Limits
Operational
Extended
Ultimate
Max Grid Cells Per Surface Field: 100000
Max Met Horizontal Array Size: max
Max Number of Puffs: 60000

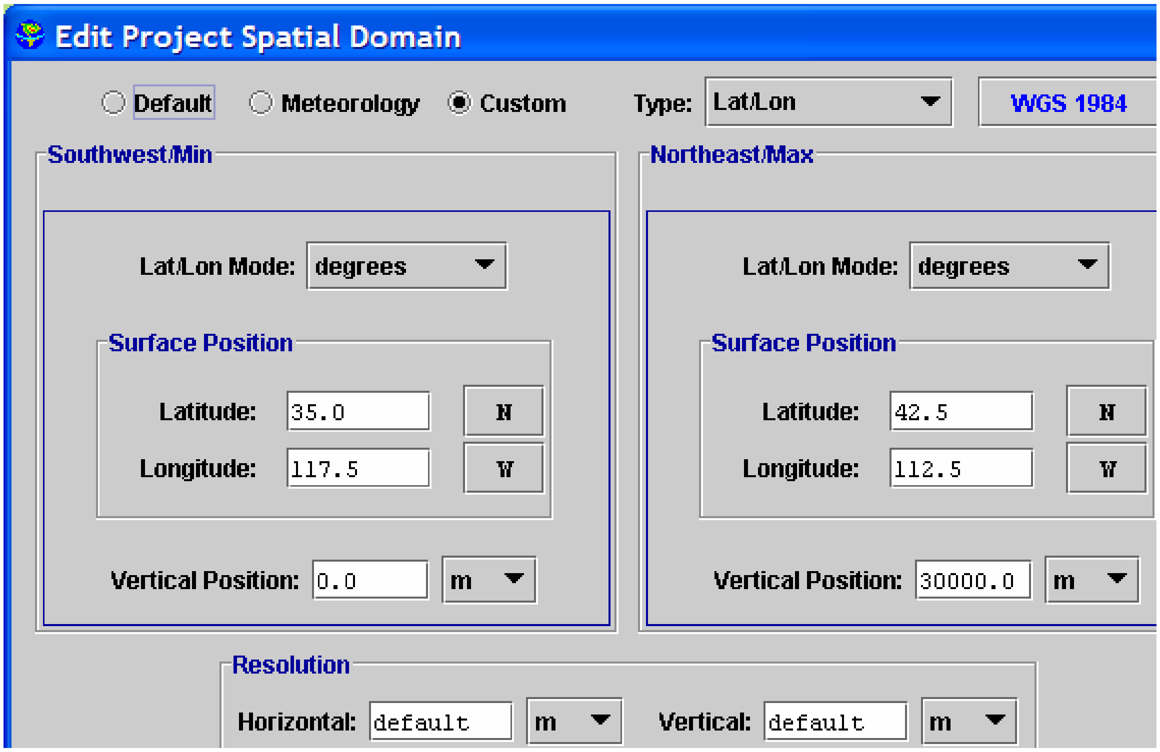

Edit Project Options
Parameters
Resolution
Samplers
Calm
Stable
Surface Dosage Height: 0.9144 m
Minimum Puff Mass: 1.0E-20 material units
Conditional Averaging: default sec
Substrate: Impermeable

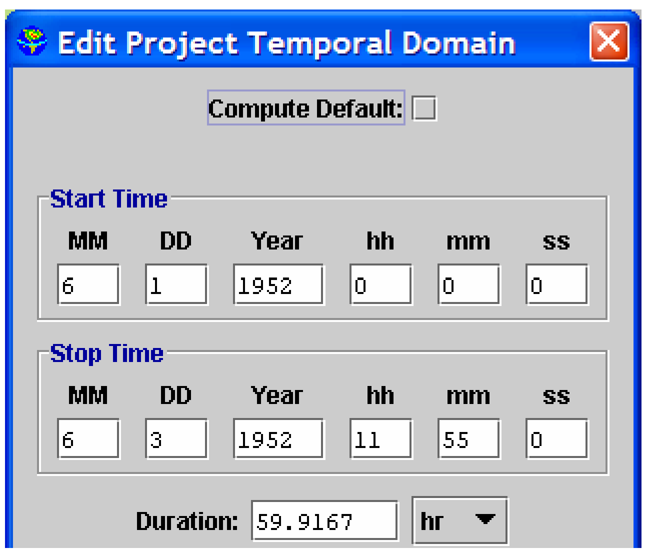

Edit Project Options
Parameters
Resolution
Samplers
Calm
Stable
Puff Split Grid Level:
2
Surface Resolution:
default
m
Boundary Layer Points:
11

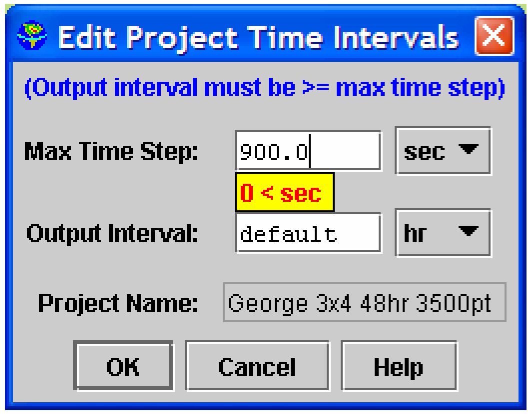

Edit Project Options
Parameters
Resolution
Samplers
Calm
Stable
Min Output Interval:
default
sec
Load Sampler File

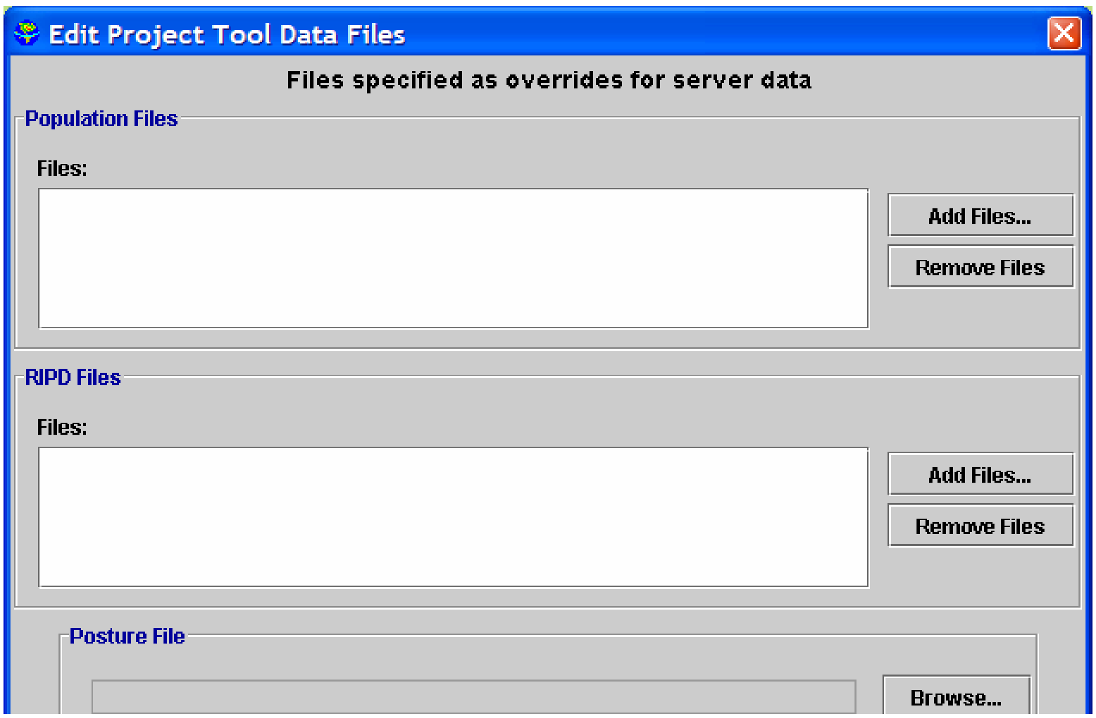

Edit Project Options
Parameters
Resolution
Samplers
Calm
Stable
Turbulence:
0.25
m2/s2
Scale:
1000.0
m

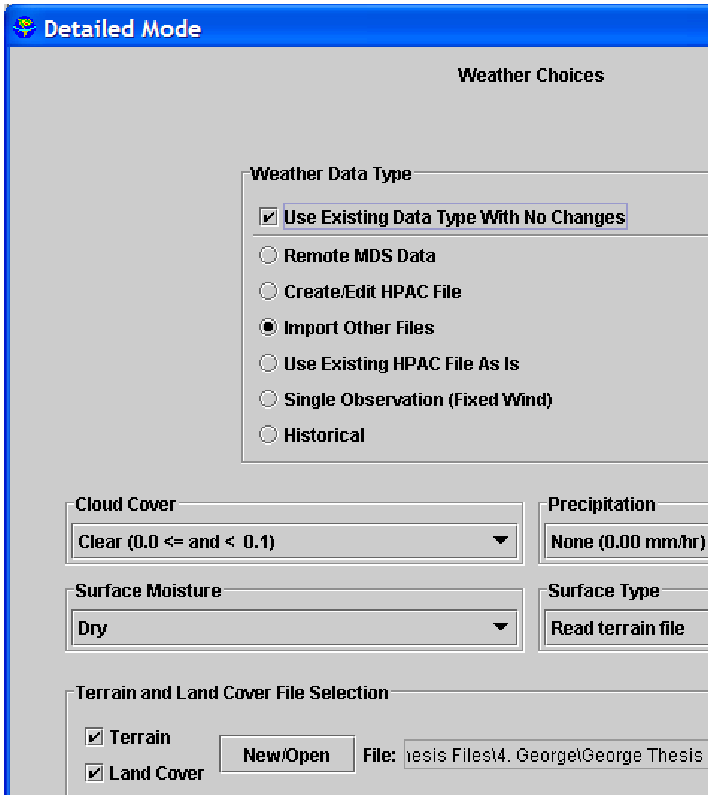

Edit Project Options
Parameters
Resolution
Samplers
Calm
Stable
Turbulence:
0.01
m2/s2
Scale:
10.0
m
Dissipation:
4.0E-4
m2/s3

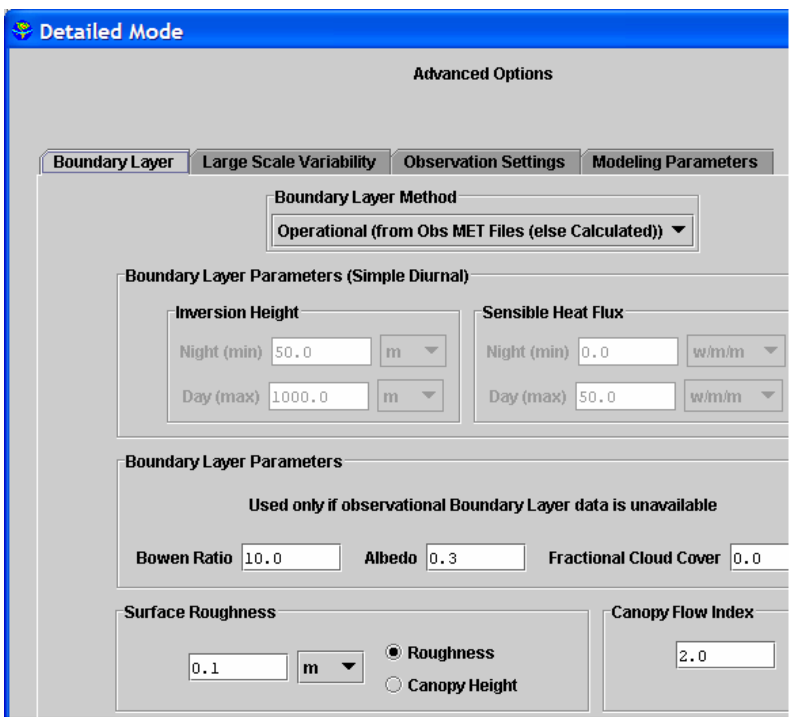

Edit Project Spatial Domain
Default
Meteorology
Custom
Type:
Lat/Lon
WGS 1984
Southwest/Min
Lat/Lon Mode:
degrees
Surface Position
Latitude:
35.0
N
Longitude:
117.5
W
Vertical Position:
0.0
m
Northeast/Max
Lat/Lon Mode:
degrees
Surface Position
Latitude:
42.5
N
Longitude:
112.5
W
Vertical Position:
30000.0
m
Resolution
Horizontal:
default
m
Vertical:
default
m

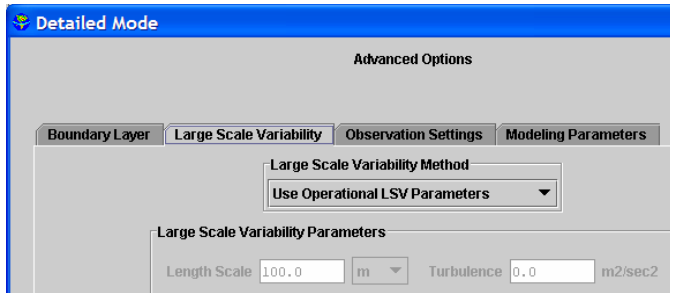

Edit Project Temporal Domain
Compute Default:
Start Time
MM DD Year hh mm ss
6 1 1952 0 0 0
Stop Time
MM DD Year hh mm ss
6 3 1952 11 55 0
Duration:
59.9167
hr

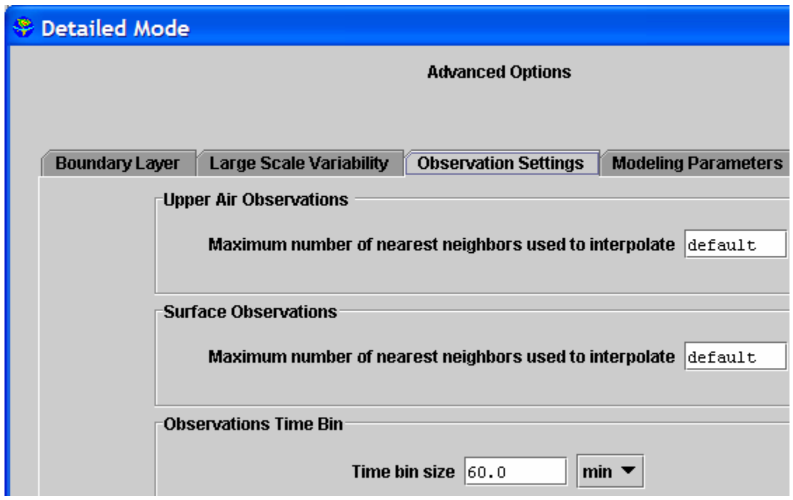

Edit Project Time Intervals
(Output interval must be >= max time step)
Max Time Step:
900.0
sec
0 < sec
Output Interval:
default
hr
Project Name:
George 3x4 48hr 3500pt
OK
Cancel
Help

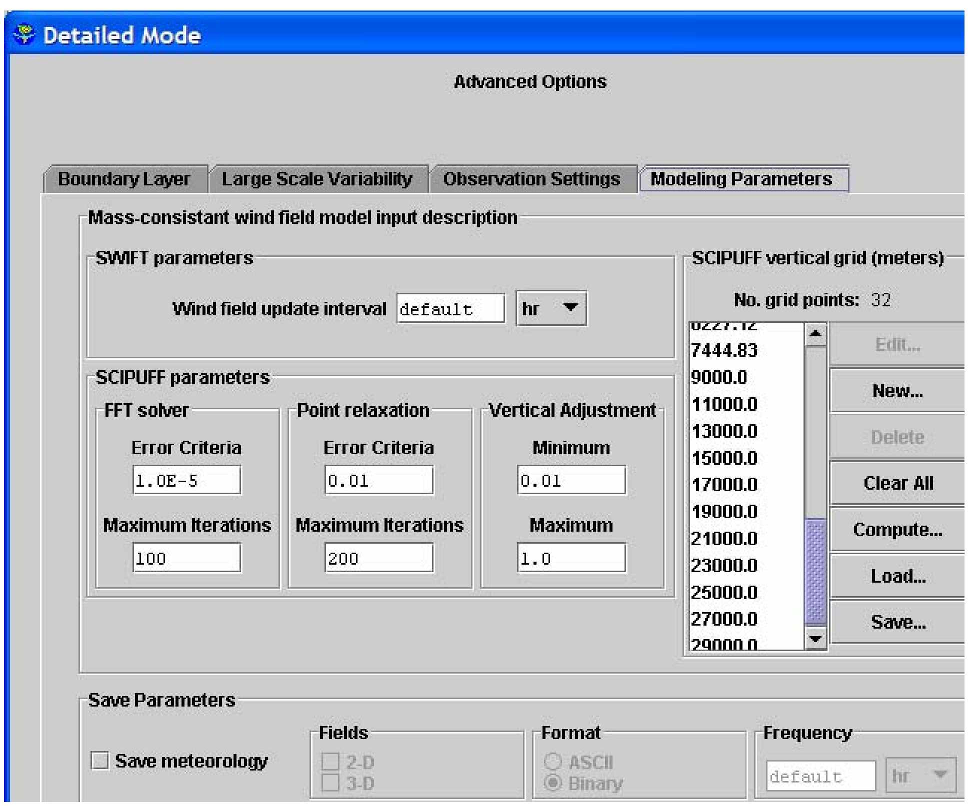

Edit Project Tool Data Files
Files specified as overrides for server data
Population Files
Files:
Add Files...
Remove Files
RIPD Files
Files:
Add Files...
Remove Files
Posture File
Browse...

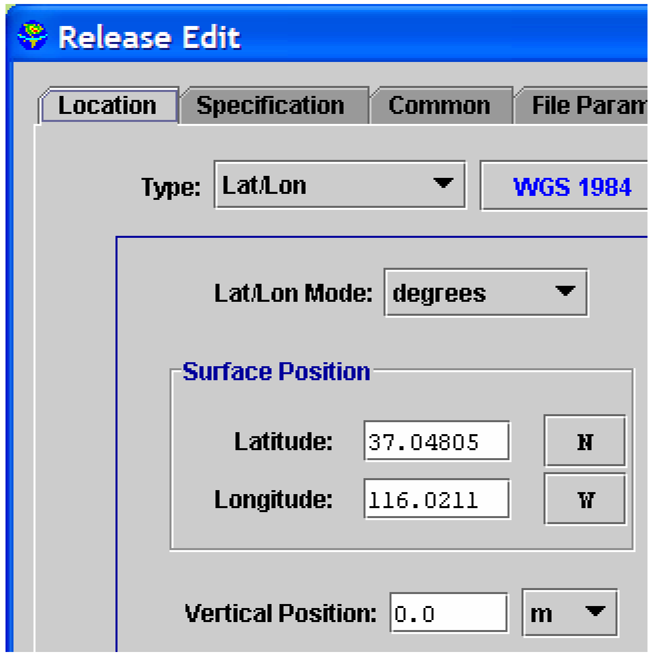

Detailed Mode
Weather Choices
Weather Data Type
Use Existing Data Type With No Changes
Remote MDS Data
Create/Edit HPAC File
Import Other Files
Use Existing HPAC File As Is
Single Observation (Fixed Wind)
Historical
Cloud Cover
Clear (0.0 <= and < 0.1)
Precipitation
None (0.00 mm/hr)
Surface Moisture
Dry
Surface Type
Read terrain file
Terrain and Land Cover File Selection
Terrain
Land Cover
New/Open
File: iesis Files\4. George\George Thesis

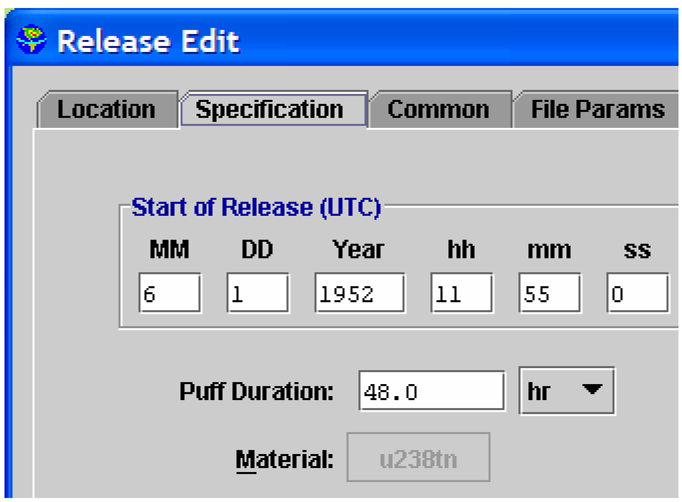

Detailed Mode
Advanced Options
Boundary Layer
Large Scale Variability
Observation Settings
Modeling Parameters
Boundary Layer Method
Operational (from Obs MET Files (else Calculated))
Boundary Layer Parameters (Simple Diurnal)
Inversion Height
Night (min) 50.0 m
Day (max) 1000.0 m
Sensible Heat Flux
Night (min) 0.0 w/m/m
Day (max) 50.0 w/m/m
Boundary Layer Parameters
Used only if observational Boundary Layer data is unavailable
Bowen Ratio 10.0
Albedo 0.3
Fractional Cloud Cover 0.0
Surface Roughness
0.1 m
Roughness
Canopy Height
Canopy Flow Index
2.0

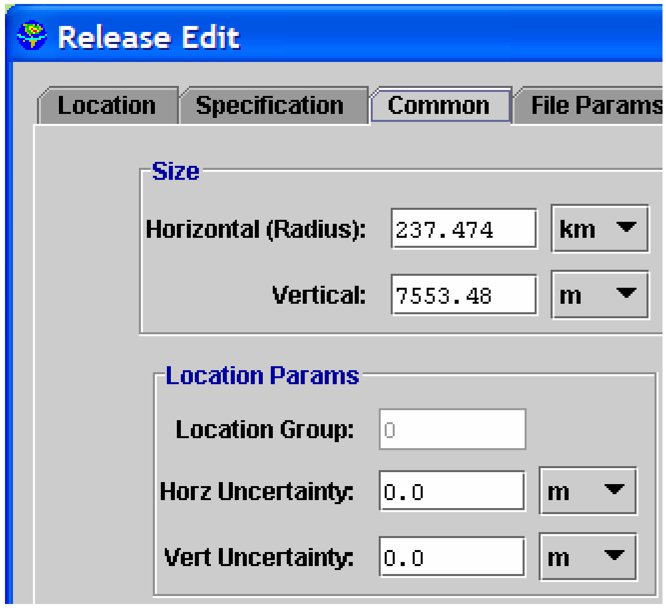

Detailed Mode
Advanced Options
Boundary Layer
Large Scale Variability
Observation Settings
Modeling Parameters
Large Scale Variability Method
Use Operational LSV Parameters
Large Scale Variability Parameters
Length Scale
100.0
m
Turbulence
0.0
m2/sec2

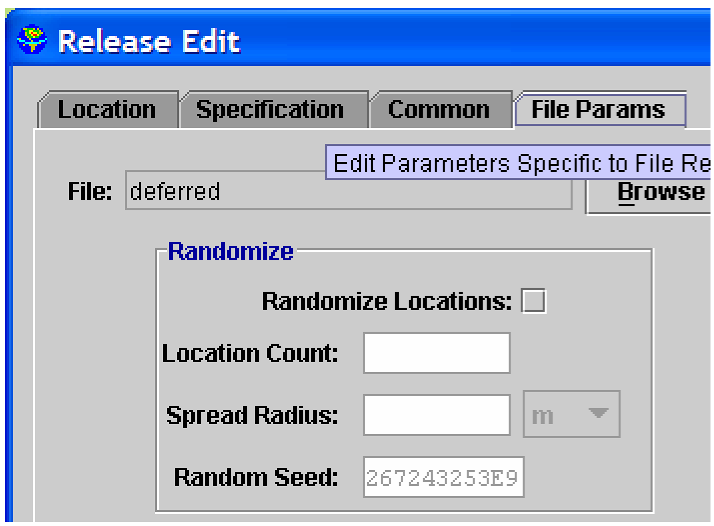

Detailed Mode
Advanced Options
Boundary Layer
Large Scale Variability
Observation Settings
Modeling Parameters
Upper Air Observations
Maximum number of nearest neighbors used to interpolate
default
Surface Observations
Maximum number of nearest neighbors used to interpolate
default
Observations Time Bin
Time bin size
60.0
min


Detailed Mode
Advanced Options
Boundary Layer
Large Scale Variability
Observation Settings
Modeling Parameters
Mass-consistant wind field model input description
SWIFT parameters
Wind field update interval
default
hr
SCIPUFF vertical grid (meters)
No. grid points: 32
7444.83
9000.0
11000.0
13000.0
15000.0
17000.0
19000.0
21000.0
23000.0
25000.0
27000.0
Edit...
New...
Delete
Clear All
Compute...
Load...
Save...
SCIPUFF parameters
FFT solver
Point relaxation
Vertical Adjustment
Error Criteria
1.0E-5
Error Criteria
0.01
Minimum
0.01
Maximum Iterations
100
Maximum Iterations
200
Maximum
1.0
Save Parameters
Save meteorology
Fields
2-D
3-D
Format
ASCII
Binary
Frequency
default
hr


Release Edit
Location
Specification
Common
File Param
Type:
Lat/Lon
WGS 1984
Lat/Lon Mode:
degrees
Surface Position
Latitude:
37.04805
N
Longitude:
116.0211
W
Vertical Position:
0.0
m


Release Edit
Location
Specification
Common
File Params
Start of Release (UTC)
MM
DD
Year
hh
mm
ss
6
1
1952
11
55
0
Puff Duration:
48.0
hr
Material:
u238tn


Release Edit
Location
Specification
Common
File Params
Size
Horizontal (Radius):
237.474
km
Vertical:
7553.48
m
Location Params
Location Group:
0
Horz Uncertainty:
0.0
m
Vert Uncertainty:
0.0
m


Release Edit
Location
Specification
Common
File Params
Edit Parameters Specific to File Re
File:
deferred
Browse
Randomize
Randomize Locations:
Location Count:
Spread Radius:
m
Random Seed:
267243253E9

Appendix E: MOE/NAD Computation FORTRAN Utility

This appendix contains the FORTRAN source code for a utility that takes a number matrix (Canvas Software Exported file) and an HPAC data file of exported dose rates and compares the two files numerically to compute MOE coordinate values and NAD values. Many remarked out lines of code were used in debugging and were left in case modifications were made that required more debugging. The source code is broken down into a main program and seven modules; GRIB2HPAC.f90, Kinds.f90, Global.f90, LocationArray.f90, TimeArray.f90, LevelArray.f90, WxArray.f90, HPAC_PRF_WRITER.f90, and Surface Elevation.dat

MatrixComparison.f90

```
Program MatrixComparison
!*********************************************************************************************
!       Purpose
!       This program will perform a comparison of 2 files.  Each file contains a list of values
!       which represent dose rates at locations on a gridded map.  The first file (from HPAC
!       Export Utility) contains the values in a single column.  These values must be arranged in
!       a 2D matrix so as to be compatible witht the values from DASA (text file is a 2D array of
!       numbers).  The values must be arranged in such a way that the position of each value in
!       the array corresponds to the same spatial location as the value in the other list.
!       For example, the first value in each list might be the most NE position on the map.
!       Also, each list must have the same number of values.  With this information, this
!       program will compute a Measure of Effectiveness (MOE)
!
!       Date                  Programmers                           Description of Change
!       ====                  ==========                            =====================
!       16 NOV 05             MAJ Kevin Pace                Original Code
!
!*********************************************************************************************
Use Kinds
Use Globals
Use Get_DASA_Utility
Use Get_HPAC_Utility
Use Array_Utility
Use MOEtools
Use Visualizer

Implicit None

Real(dp) ::DASA2D(-900:900, -900:900) ! Array of values for DASA Values (3600 square miles)
Real(dp) ::HPAC2D(-900:900, -900:900) ! Array of values for HPAC Values (3600 square miles)
Integer  ::Top, Bottom, Left, Right          ! Row/Column Array Boundaries for trimmed array
Real(dp) :: MOEx(1:7), MOEy(1:7)             ! X and Y coordinates for Measure of Effectiveness
Integer  :: NmbrCntrs                        ! Number of contour intervals to be compared

DASA2D = -2.0_dp ! Initialize arrays to something recognizable in case of error in filling them
up
HPAC2D = -3.0_dp

Call FillDASA(DASA2D)  ! Fills the DASA array with values from Canvas-created text file

Call FillHPAC(HPAC2D, NmbrCntrs)       ! Fills the HPAC array with values from HPAC-exported
                                       ! text file

Call TrimArray(DASA2D, HPAC2D, Top, Bottom, Left, Right)

Call
CalculateStats(DASA2D(Top:Bottom,Left:Right),HPAC2D(Top:Bottom,Left:Right),Top,Bottom,Left,Right,
NmbrCntrs)
```

```
!Dasa2d = Dasa2d + Hpac2d
! When calling Visualizer you must look at the plot from below to see the plot match DASA/HPAC
Call Array_Visualizer (DASA2D(Top:Bottom, Left:Right) , Top, Bottom, Left, Right)
Call Array_Visualizer (HPAC2D(Top:Bottom, Left:Right), Top, Bottom, Left, Right)

End Program MatrixComparison
```

Kinds.f90

```
Module Kinds

       Implicit None
       Public

!************************************************************************************************
!      Date                  Programmers                          Description of Change
!      ====                  ==========                           =====================
!      24 Jan 05             MAJ Kevin Pace               Original Code
!************************************************************************************************

Integer,Parameter:: sp = Selected_Real_Kind(p=6)
Integer,Parameter:: dp = Selected_Real_Kind(p=14)

End Module Kinds
```

Globals.f90

```
Module Globals

Use Kinds
Implicit None

Integer:: i, j, x, y                   ! Loop Counters
Character(len=50):: HPAC, DASA         ! Files that contain dose rate values
Integer:: IOSTAT                       ! Error handler for reading text files
Integer:: ierror1                      ! Error handler for opening text files
Integer  :: HPACx, HPACy               ! # of x and y points exported from HPAC
Integer, Parameter  :: GZ = 0 ! Pixel value for the location of (Ground Zero) GZ in DASA values
Integer, Parameter  :: GZplus20N = 1  ! Pixel value for location 20 miles directly N of GZ
Real(dp), Parameter :: DR0 = 255._dp  ! Value for dose rates below threshold contour level value
Real(dp), Parameter :: DR1 = 225._dp  ! lowest contour level value (threshold level value)
Real(dp), Parameter :: DR2 = 200._dp  ! Value for 2nd lowest contour level value (if present)
Real(dp), Parameter :: DR3 = 175._dp  ! Value for 3rd lowest contour level value (if present)
Real(dp), Parameter :: DR4 = 130._dp  ! Value for 4th lowest contour level value (if present)
Real(dp), Parameter :: DR5 =  90._dp  ! Value for 5th lowest contour level value (if present)
Real(dp), Parameter :: DR6 =  45._dp  ! Value for 6th lowest contour level value (if present)
Real(dp), Parameter :: DR7 =  20._dp  ! Value for 7th lowest contour level value (if present)
Real(dp), Parameter :: Contour(1:7) = (/DR1, DR2, DR3, DR4, DR5, DR6, DR7/)
End Module Global
```

DASA_Utility.f90

```
Module Get_DASA_Utility

Use Kinds
Use Globals
Implicit None

Contains

Subroutine FillDASA (DASA2D)
!************************************************************************************************
! Fills the DASA array with values.  The text file's origin is from a Canvas export of values.
!
!************************************************************************************************

Use Kinds
Use Globals
Use Visualizer
Implicit None

Real(dp), Intent(InOut) :: DASA2D(-900:900,-900:900) ! 2D array for holding dose rate values
Real(dp), Allocatable:: Temp2D(:,:)
Character(len=50):: DASA               ! File containing Canvas exported values
Integer :: xsize, ysize                ! X and Y dimensions of DASA file
```

```
Integer :: value              ! Value of pixel
Integer :: Size1(1:1E6)       ! This array will hold every value in the file.
                              ! Allows an area of about 1E7 square miles (3Kx3K or 2Kx5K, etc)

Integer :: Elements           ! Number of elements in my values matrix
Integer :: StartColumn, EndColumn
Integer :: GZx, GZy, GZNorthx, GZNorthy
Integer :: TLx, TLy           ! (1,1) position of Temp2D with reference to GZ

!Initialize Variables
ierror1= 0
xsize  = 0
ysize  = 0
IOSTAT = 0
Size1 = -3

Write(*,*) 'Please enter filename containing DASA Values (Observation): ' !  Obtain File Name
Read(*,*) DASA

!********************Open File
OPEN (UNIT = 20, FILE=DASA, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'Error Opening DASA file'

!********************Find out dimensions (1 value = 1 pixel, 3 pixels = 1 mile) of the value
matrix (x-size and y-size)
Do ! Find out how many rows are in the file.  This is ysize.

       Read (20,*, IOSTAT = ierror1)        ! Read line

       If (ierror1 .NE. 0) Then
              Write(*,*) "There are ",ysize, " lines in the DASA value matrix text file"
              ierror1 = 0
              Exit
       End If

       ysize = ysize+1  !Amount of rows

End Do
Write(*,*) "Ysize =", ysize
Rewind(20)  !Take read pointer to beginning of file


! Find out how many columns are in the file
Read (20,*, IOSTAT = ierror1) Size1(:) ! Fill Size array with every number in value matrix
Rewind(20)  !Take read pointer to beginning of file

Do i = 1,1E6 ! Go through array until -1 (there are no negative numbers in these value files
       If (Size1(i) .EQ. -3) Then
              Elements = i-1
              xsize = elements/ysize  !(# values in text file)/(number of rows)=# of columns.
              Write(*,*) "xsize = ", xsize  !Amount of columns
              Exit
       End If
End Do

Close(20)

!***********Fill Temporary 2D Array
Allocate (Temp2D(1:ysize,1:xsize))
Temp2D = 255

Do y = 1, ysize       !Number of rows

       StartColumn = (xsize*(y-1))+1
       EndColumn   =   xsize*y
       Temp2D(y, :) = Real(Size1(StartColumn:EndColumn),dp)

End Do

!**********Find Position of GZ and GZ + 20 miles North
GZx = -5
GZy = -5
GZNorthx = -5
GZNorthy = -5

Do y = 1, ysize  !Searches through Temp2D until it finds BOTH GZ and GZNorth
       Do x = 1, xsize
              If(Temp2D(y,x) .EQ. 1._dp) Then
                     GZNorthx = x
                     GZNorthy = y
              End If
```

```
        If(Temp2D(y,x) .EQ. 0._dp) Then
            GZx = x
            GZy = y
        End If

        If (GZx .NE. -5 .AND. GZNorthx .NE. -5) Exit !find both values before search stops
    End Do
End Do

Write(*,*) "GZ coordinates are: ", GZx, GZy
Write(*,*) "GZNorth coordinates are: ", GZNorthx, GZNorthy

!************ Insert Values of Temp2D onto DASA2D with GZ = (0,0) and GZNorth = (0,60)

! Find origin of Temp2D with reference to GZ
TLx = -GZx +1 ! Add 1 because matrix starts at (1,1), NOT (0,0)
TLy = -GZy +1

DASA2D = 255 ! Dose rate of zero is 255 for DASA files
DASA2D(TLy:(ySize-GZy), TLx:(xSize-GZx)) = Temp2D(1:ysize, 1:xsize)

DeAllocate(Temp2D)

!************ Filter values to ensure they are all a specific contour level and not in between
Call FilterDASA2D (DASA2D)


End Subroutine FillDASA
!**************************************************


Subroutine FilterDASA2D (DASA2D)
!****************************************************************************************************
!This forces all array values (pixel values) into a bin; either a 255, 225, 200, 175, 130, 90, 45
!or 20.  These values are reserved for dose rates.
!0 roentgens/hr = 255 (White on grayscale values; 0 = black and 255 = white)
!Lowest Dose-Rate =225 (Very Light Gray)
!Second Lowest Dose-Rate = 200
!Third Lowest Dose-Rate = 175
!And so on until .....
!7th lowest dose rate (which has to be the highest dose rate for this program) = 20(almost Black)
!****************************************************************************************************

Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut) :: DASA2D(-900:900, -900:900)! # of values contained in HPAC value field
Integer :: A, B, C, D, E, F, G, H            ! Dose Rate bins A=225, B=200, C=175, etc

Do y = -900, 900
Do x = -900, 900
    If (DASA2D(x,y) .GT. (DR0+DR1)/2._dp) Then !Greater than half way between 225 and 255
        DASA2D(x,y) = DR0
    Else if (DASA2D(x,y) .LE. (DR0+DR1)/2._dp .AND. DASA2D(x,y) .GT. (DR1+DR2)/2._dp) Then
        DASA2D(x,y) = DR1
    Else if (DASA2D(x,y) .LE. (DR1+DR2)/2._dp .AND. DASA2D(x,y) .GT. (DR2+DR3)/2._dp) Then
        DASA2D(x,y) = DR2
    Else if (DASA2D(x,y) .LE. (DR2+DR3)/2._dp .AND. DASA2D(x,y) .GT. (DR3+DR4)/2._dp) Then
        DASA2D(x,y) = DR3
    Else if (DASA2D(x,y) .LE. (DR3+DR4)/2._dp .AND. DASA2D(x,y) .GT. (DR4+DR5)/2._dp) Then
        DASA2D(x,y) = DR4
    Else if (DASA2D(x,y) .LE. (DR4+DR5)/2._dp .AND. DASA2D(x,y) .GT. (DR5+DR6)/2._dp) Then
        DASA2D(x,y) = DR5
    Else if (DASA2D(x,y) .LE. (DR5+DR6)/2._dp .AND. DASA2D(x,y) .GT. (DR6+DR7)/2._dp) Then
        DASA2D(x,y) = DR6
    Else if (DASA2D(x,y) .LE. (DR6+DR7)/2._dp .AND. DASA2D(x,y) .GT. (DR7 - 10._dp)) Then
        DASA2D(x,y) = DR7
    Else if (DASA2D(x,y) .EQ. 1._dp) Then
        DASA2D(x,y) = 1._dp
    Else if (DASA2D(x,y) .EQ. 0._dp) Then
        DASA2D(x,y) = 0._dp
    Else
        Write(*,*) x, y, DASA2D(x,y)
    End If
End Do
End Do

End Subroutine FilterDASA2D
```

```
End Module Get_DASA_Utility
```

HPAC_Utility.f90

```
Module Get_HPAC_Utility

Use Kinds
Use Globals
Implicit None

Contains

Subroutine FillHPAC(HPAC2D, NmbrCntrs)
!*************************************************************************************************
!Fills the DASA array with values.  The text file's origin is from a Canvas export of values.  As
!an intermediate step, the array is set to values at the actual dose rates.  The final product,
!however, is a matrix of terraced values where 0 roentgens/hr = 255 (White on grayscale) and 0 is
!black.  This is the same scale that DASA2D is in and it is easier (at least it makes more sense)
!to get HPAC into the grayscale than DASA into the actual contour level.
!*************************************************************************************************

Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut) :: HPAC2D(-900:900,-900:900)! 2D array for holding dose rate values
Character(len=50):: HPAC              ! File containing Canvas exported values
Integer :: HPACrows                   ! # of data points in HPAC value file
Integer :: xsize, ysize               ! X and Y dimensions of DASA file
Real(dp), Allocatable:: Value(:)      ! 1D Arrays of dose-rate values
Real(dp), Allocatable:: Temp2D(:,:)   ! 2D array of values, this will used to form HPAC2D
!Integer :: GZx, GZy, GZNorthx, GZNorthy
!Integer :: TLx, TLy                  ! (1,1) position of Temp2D with reference to GZ
Integer, Intent(InOut) :: NmbrCntrs   ! Number of Contour level values

!Initialize Variables
ierror1 =  0
xSize   =  0
ySize   =  0

!********************Get HPAC Value File Name
Write(*,*) 'Please enter filename containing HPAC Values (Prediction): ' !  Obtain File Name
Read(*,*) HPAC


!********************Find out How many values are contained in the HPAC value file
Call GetHPACsize (HPACrows, HPAC)


!********************Load HPAC dose-rates into 1D array; return dimensions of Temp 2D array
!(which will become HPAC2D)
Allocate(Value(1:HPACrows))
Value = -9999._dp
Call GetHPACValues (ySize, xSize, Value, HPAC)


!********************Terrace the values to match HPAC output
Call NormalizeValues(Value, NmbrCntrs)


!**********************Fill Temp2D Array with values
Allocate (Temp2D(1:ySize, 1:xSize))   ! Allocate Temporary 2D array (which will become HPAC2D)
Temp2D = -9999                        ! Initialize Temp2D array to erroneous value
Call FillTemp2D (Temp2D, value, xSize, ySize)
Deallocate(Value)                     ! No longer needed as the values are in 2D form now


!**********************Fill HPAC2D with values [GZ at (0,0), HPAC data is always oriented N!]
Call FillHPAC2D (HPAC2D, Temp2D)
Deallocate(Temp2D)                    ! No longer needed

End Subroutine FillHPAC
!**************************************************


Subroutine GetHPACsize (HPACrows, HPAC)
!*************************************************************************************************
! Opens the HPAC value file and reads the # of rows that contain dose-rate values
```

```
!*************************************************************************************************

Use Kinds
Use Globals
Implicit None

Integer, Intent(InOut) :: HPACrows    ! # of values contained in HPAC value fiel
Character(Len=50), Intent(In):: HPAC  ! Name of the Exported Value file from HPAC
Character(Len=1):: HeaderFlag         ! First character in a line

ierror1 = 0

! Open HPAC value File
OPEN (UNIT = 30, FILE=HPAC, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'HPAC_Utility, GetHPACsize:  Error Opening HPAC value file'

!Read over header files and get ready to read first data point of HPAC file
DO
       Read(30, *, IOSTAT = ierror1) HeaderFlag
       If (ierror1 .NE. 0) Then
              Write(*,*) "HPAC_Utility, GetHPACsize: File empty or contained no values"
              Exit
       End If

       If (HeaderFlag .NE. '#') Then
              Backspace(30)
              Exit
       End If
End Do

!Count how many rows of data points there are in this file
Do
       Read (30, *, IOSTAT = ierror1)       ! Read a line

       If (ierror1 .NE. 0) Then
              Write(*,*) "HPAC_Utility, GetHPACsize:  EOR is found"
              Write(*,*) "This file has ", HPACrows, " rows"
              ierror1 = 0
              Exit
       End If

       HPACrows = HPACrows + 1
End Do

Close (30)

End Subroutine GetHPACsize
!**************************************************


Subroutine GetHPACValues (ySize, xSize, Value, HPAC)
!*************************************************************************************************
! Opens the HPAC value file and
!
!*************************************************************************************************

Use Kinds
Use Globals
Implicit None

Integer, Intent(InOut):: ySize, xSize ! dimensions of geographical area covered by HPAC file
Real(dp), Intent(InOut):: Value(:)    ! 1D array of dose-rate values
Character(Len=50), Intent(In):: HPAC  ! Name of the Exported Value file from HPAC
Real(dp), Allocatable :: Lon(:)    ! Array of longitudes
Character(Len=1)     :: HeaderFlag ! First character in a line
Character(Len=100)   :: Line    ! Entire line from HPAC value file
Integer              :: Arrow   ! Pointer when reading through LINE
Real(dp)             :: Lon2Look4 !Flag for detecting a repeating lon, used for determining xSize

ierror1 = 0

Allocate(Lon(1:Size(Value)))
Lon = -9999._dp

! Open HPAC value File
OPEN (UNIT = 30, FILE=HPAC, STATUS='OLD', ACTION='READ', IOSTAT=ierror1)
If (ierror1 .NE. 0) Write(*,*) 'HPAC_Utility, GetHPACvalues:  Error Opening HPAC value file'

!Read over header files and get ready to read first data point of HPAC file
DO
       Read(30, *, IOSTAT = ierror1) HeaderFlag
```

```
        If (ierror1 .NE. 0) Then
               Write(*,*) "HPAC_Utility, GetHPACvalues: File empty or contained no values"
               Exit
        End If

        If (HeaderFlag .NE. '#') Then
               Backspace(30)  !Backup a row and get ready to read values
               Exit
        End If
End Do

! Reads every non-header line and extracts longitude and dose-rate (in rads/hr)
Do  i = 1, Size(Value)
        Read (30,'(a)', IOSTAT = ierror1) LINE       ! Read a line

        If (ierror1 .NE. 0) Then
               !Write(*,*) "HPAC Utility:  EOR is found"
               ierror1 = 0
               Exit
        End If

        arrow = index(Line,"(") + 1           ! Find the ( before the Longitude
        Read (Line(arrow:),*) Lon(i)          ! Read the Longitude (Repeats when a Row Changes)
        arrow = index(Line,")") + 1           ! Find the ) before the Dose Rate
        Read (Line(arrow:),*) Value(i)        ! Read the Dose Rate (Rads/hour)

End Do

Lon2Look4 = Lon(1) ! This sets Lon2Look4 as flag for detecting when the lon starts repeating

Do i = 2, Size(Value)  !Start looking for repeating lon
                       !(Lon(1) is Lon2Look4 so we start at Lon(2))
        If (Lon(i) .EQ. Lon2Look4) Then
               xSize = i-1
               Exit
        Else If (i .EQ. Size(Value) .AND. xSize .EQ. 0) Then
               Write(*,*) "HPAC_Utility, GetHPACvalues:  The Longitude never repeated"
        Exit
        End If
End Do

Write(*,*) "There are ", xSize, " columns in our future HPAC2D array"

! Find ySize by doing some division
ySize = (Size(value))/xSize
Write(*,*) "There are ", ysize,    " rows in our future HPAC2D array"

Deallocate(Lon)
Close (30)

End Subroutine GetHPACvalues
!**************************************************



Subroutine NormalizeValues (Value, NmbrCntrs)
!**************************************************************************************************
!  This subroutine finds out what contours are of interest in this comparison.
!  Prompts the user for the number of contours of interest (up to 10)
!  Then it asks for the time of interest (in hours after the detonation) for normalization to H+1
!  Finally a calculation is made to turn HPAC rads/hr at H+'many' into DASA roentgens/hr at H+1
!  This module then turns these 'continuous' values at H+1 into a step function of values.
!  This should produce a 'terraced' array with values at contours of interest and zero.  (11 !
!  values max)
!**************************************************************************************************

Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut)       :: Value(:)    ! 1D Array of values from HPAC value file
Integer, Intent(InOut)  :: NmbrCntrs         ! Number of Contours for comparison
Real(dp)                      :: ContourData(1:7,1:2) ! DASA Contour Value, Hours after
                                                !detonation in HPAC, HPAC rads/hr @ H+1
Character(Len=1)    :: Answer
Real(dp)            :: HPACtime              ! How long HPAC ran (post H-hour) before it made a
calculation
Logical             :: NoSwap                ! Used in sort
Real(dp)            :: Dummy(1:2)            ! Used in sort

! Get Number of DASA contours
```

```
Do
       Write(*,*) 'Please enter the number of contours (from DASA test data) for comparison: '
       Read(*,*) NmbrCntrs
       Write(*,50) NmbrCntrs
       50 Format ("You want to compare ", i2, " contours.")
       Write(*,*) "Is this correct? (Y/N)"
       Read(*,*) Answer
       If (Answer .EQ. "Y" .OR. Answer .EQ. "y") Then
              Exit
       End If
End Do

ContourData = 0.0_dp

! Gets the DASA contour values in roentgen/hr
Do
       Do i = 1, NmbrCntrs ! Get Contour Values
              Write(*,100) i
              100 Format ("Enter contour value (roentgens/hr) #", i2, ":")  !
              Write(*,*) "Enter -9999 to Exit"
              Read(*,*) ContourData(i,1)
              If (ContourData(i,1) .EQ. -9999) Exit !Exit opportunity
       End Do

       If (MinVal(ContourData) .EQ. -9999._dp) Exit  !Continuation of Exit Opportunity

       ContourData(NmbrCntrs+1:7, 1) = MaxVal(ContourData(:,1))

       Do i = 1, 7
              Write(*,200) i, ContourData(i,1)
              200 Format ("Contour # ", i3," has a value of ", E9.3)
       End Do

       Write(*,*) "Is this correct? (Y/N)"
       Read(*,*) Answer
       If (Answer .EQ. "Y" .OR. Answer .EQ. "y") Exit
End Do


!Obtain t for t^-1.3 calculation
!Write(*,*) 'Please how long HPAC ran (in hours) post detonation before making a calculation: '
!Read(*,*) HPACtime
HPACtime = 48._dp

!Sort the ContourData array to ensure the smallest dose rate is first
Do i = 1, 6
       NoSwap = .TRUE.
       Do j = 7, i+1, -1
              If (ContourData(j, 1) .LT. ContourData(j-1, 1)) Then
                     NoSwap = .False.
                     Dummy(:) = ContourData(j, :)
                     ContourData(j, :) = ContourData(j-1, :)
                     ContourData(j-1, :)  = Dummy(:)
              End If
       End Do
       If (NoSwap) Exit
End Do

!Find DASA to HPAC dose rate conversion
!HPAC uses t^-1.3 instead of t^-1.2 and a .7 (roentgen to Rad conversion)
!The third column of the ContourData array are DASA-equivalent HPAC values (in rad/hr)
!These are the levels of our step-function.  (Basically bins for values to fall into)
ContourData(:, 2) = ContourData(:,1) * (HPACtime**(-1.3_dp)) * 0.7_dp

Write(*,*) "Sorted from Lowest to Highest"
Do i = 1, 7
       Write (*,500) ContourData(i, 1), ContourData(i, 2)
       500 Format ("DASA Dose Rate: ", E10.5, "Normalized to H+1: ", E10.5)
End Do

!Convert values in Value array into either zero, or an H+1 normalized contour level value
Do i = 1, Size(Value)
       If(Value(i) .LT. ContourData(1, 2)) Then     ! Is it lower than the lowest
              Value(i) = 0._dp
       Else If(Value(i) .GE. ContourData(NmbrCntrs, 2)) Then ! Is it higher than the highest
              Value(i) = ContourData(NmbrCntrs, 2)
       Else
              Do j = 1, NmbrCntrs - 1
                If(Value(i) .GE. ContourData(j, 2) .AND. Value(i) .LT. ContourData(j+1, 2)) Then
                     Value(i) = ContourData(j, 2)
                     Exit
```

```
                End If
              End Do
       End If
End Do

Call FilterValues(Value, ContourData, NmbrCntrs)

End Subroutine NormalizeValues
!**************************************************


Subroutine FilterValues (Value, ContourData, NmbrCntrs)
!*************************************************************************************************
!This forces all array values (pixel values) into a bin; either a 255, 225, 200, 175, 130, 90, 45
!or 20.  These values are reserved for dose rates.
!0 roentgens/hr = 255 (White on grayscale values; 0 = black and 255 = white)
!Lowest Dose-Rate =225 (Very Light Gray)
!Second Lowest Dose-Rate = 200
!Third Lowest Dose-Rate = 175
!And so on until .....
!7th lowest dose rate (which has to be the highest dose rate for this program)= 20 (almost Black)
!*************************************************************************************************
Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut) :: Value(:) !(Terraced Dose Rates in, Terraced Grayscale values out)
Real(dp), Intent(In)    :: ContourData(1:7, 1:2)! # of values contained in HPAC value field
Integer, Intent(In)     :: NmbrCntrs


Select Case (NmbrCntrs)
  Case(1)
       Do i = 1, Size(Value)
              If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                     Value(i) = DR0
              Else If (Value(i) .GE. ContourData(1,2)) Then
                     Value(i) = DR1
              Else
                     Write(*,*) i, Value(i)
              End If
       End Do

  Case(2)
       Do i = 1, Size(Value)
              If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                     Value(i) = DR0
              Else If (Value(i) .GE. ContourData(1,2) .AND. Value(i) .LT. ContourData(2,2)) Then
                     Value(i) = DR1
              Else If (Value(i) .GE. ContourData(2,2)) Then
                     Value(i) = DR2
              Else
                     Write(*,*) i, Value(i)
              End If
       End Do

  Case(3)
       Do i = 1, Size(Value)
              If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                     Value(i) = DR0
              Else If (Value(i) .GE. ContourData(1,2) .AND. Value(i) .LT. ContourData(2,2)) Then
                     Value(i) = DR1
              Else If (Value(i) .GE. ContourData(2,2) .AND. Value(i) .LT. ContourData(3,2)) Then
                     Value(i) = DR2
              Else If (Value(i) .GE. ContourData(3,2)) Then
                     Value(i) = DR3
              Else
                     Write(*,*) i, Value(i)
              End If
       End Do

  Case(4)
       Do i = 1, Size(Value)
              If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                     Value(i) = DR0
              Else If (Value(i) .GE. ContourData(1,2) .AND. Value(i) .LT. ContourData(2,2)) Then
                     Value(i) = DR1
              Else If (Value(i) .GE. ContourData(2,2) .AND. Value(i) .LT. ContourData(3,2)) Then
                     Value(i) = DR2
              Else If (Value(i) .GE. ContourData(3,2) .AND. Value(i) .LT. ContourData(4,2)) Then
                     Value(i) = DR3
```

```
            Else If (Value(i) .GE. ContourData(4,2)) Then
                  Value(i) = DR4
            Else
                  Write(*,*) i, Value(i)
            End If
      End Do

  Case(5)
      Do i = 1, Size(Value)
            If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                  Value(i) = DR0
            Else If (Value(i) .GE. ContourData(1,2) .AND. Value(i) .LT. ContourData(2,2)) Then
                  Value(i) = DR1
            Else If (Value(i) .GE. ContourData(2,2) .AND. Value(i) .LT. ContourData(3,2)) Then
                  Value(i) = DR2
            Else If (Value(i) .GE. ContourData(3,2) .AND. Value(i) .LT. ContourData(4,2)) Then
                  Value(i) = DR3
            Else If (Value(i) .GE. ContourData(4,2) .AND. Value(i) .LT. ContourData(5,2)) Then
                  Value(i) = DR4
            Else If (Value(i) .GE. ContourData(5,2)) Then
                  Value(i) = DR5
            Else
                  Write(*,*) i, Value(i)
            End If
      End Do

  Case(6)
      Do i = 1, Size(Value)
            If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                  Value(i) = DR0
            Else If (Value(i) .GE. ContourData(1,2) .AND. Value(i) .LT. ContourData(2,2)) Then
                  Value(i) = DR1
            Else If (Value(i) .GE. ContourData(2,2) .AND. Value(i) .LT. ContourData(3,2)) Then
                  Value(i) = DR2
            Else If (Value(i) .GE. ContourData(3,2) .AND. Value(i) .LT. ContourData(4,2)) Then
                  Value(i) = DR3
            Else If (Value(i) .GE. ContourData(4,2) .AND. Value(i) .LT. ContourData(5,2)) Then
                  Value(i) = DR4
            Else If (Value(i) .GE. ContourData(5,2) .AND. Value(i) .LT. ContourData(6,2)) Then
                  Value(i) = DR5
            Else If (Value(i) .GE. ContourData(6,2)) Then
                  Value(i) = DR6
            Else
                  Write(*,*) i, Value(i)
            End If
      End Do

  Case(7)
      Do i = 1, Size(Value)
            If (Value(i) .LT. ContourData(1,2)) Then !Anything less than lowest contour level
                  Value(i) = DR0
            Else If (Value(i) .GE. ContourData(1,2) .AND. Value(i) .LT. ContourData(2,2)) Then
                  Value(i) = DR1
            Else If (Value(i) .GE. ContourData(2,2) .AND. Value(i) .LT. ContourData(3,2)) Then
                  Value(i) = DR2
            Else If (Value(i) .GE. ContourData(3,2) .AND. Value(i) .LT. ContourData(4,2)) Then
                  Value(i) = DR3
            Else If (Value(i) .GE. ContourData(4,2) .AND. Value(i) .LT. ContourData(5,2)) Then
                  Value(i) = DR4
            Else If (Value(i) .GE. ContourData(5,2) .AND. Value(i) .LT. ContourData(6,2)) Then
                  Value(i) = DR5
            Else If (Value(i) .GE. ContourData(6,2) .AND. Value(i) .LT. ContourData(7,2)) Then
                  Value(i) = DR6
            Else If (Value(i) .GE. ContourData(7,2)) Then
                  Value(i) = DR7
            Else
                  Write(*,*) i, Value(i)
            End If
      End Do

End Select

End Subroutine FilterValues


Subroutine FillTemp2D (Temp2D, Value, xSize, ySize)
!*******************************************************************************************
! Fills the Temporary 2D array with values.  HPAC values in the value file are in the following
! order:  Start at bottom left, end at top right, going row by row.  So its like reading except
! you start from the bottom of the page.  If the HPAC file contained 4 rows and 3 columns, the
! values would be placed in this order:
```

```
!
!  10  11  12
!  7   8   9
!  4   5   6
!  1   2   3
!  The problem is that FORTRAN Matricies are in this order:
!  1   2   3
!  4   5   6
!  7   8   9
!  10  11  12
!  So I need to 'flip' Temp2D to have the matrix geographically correct.  That is, I want the
!  last row to be the first, the second to last to be the second, etc
!*******************************************************************************************
Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut) :: Temp2D(:,:) ! Temporary 2D array that holds HPAC values
Real(dp), Intent(In)  :: Value(:)       ! 1D Array of values from HPAC value file
Integer,  Intent(In)  :: ySize, xSize !Dimensions (in points) of area covered by HPAC value file
Integer :: StartColumn, EndColumn ! Beginning and End Sequence numbers of a row of HPAC values
Real(dp), Allocatable:: FlippedArray(:,:)    ! Array used to flip Temp2D
Allocate(FlippedArray(1:ySize, 1:xSize))

Do y = 1, ySize                      !Number of rows
       StartColumn = (xsize*(y-1))+1 ! First item in the row
       EndColumn   =   xsize*y       ! Last item in the row
       Temp2D(y, :) = Value(StartColumn:EndColumn)
End Do

Do y = 1, ySize       !Create Flipped Array
       FlippedArray(y, :) = Temp2D(ySize-(y-1), :)
End Do

Temp2D = FlippedArray

Deallocate(FlippedArray)

End Subroutine FillTemp2D
!*************************************************

Subroutine FillHPAC2D (HPAC2D, Temp2D)
!*******************************************************************************************
! This subroutine places Temp2D onto the larger DASA2D with GZ at (0,0)
!*******************************************************************************************
Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut) :: HPAC2D(-900:900,-900:900) ! 2D array (HPAC output
Real(dp), Intent(In)   :: Temp2D(:,:) ! Temporary 2D array that holds HPAC values
Integer                 :: NWx, NWy   ! Miles West/South of GZ that HPAC file contains
Character(Len=1)        :: Answer     ! Answer to Yes or No questions
Integer                 :: TLx, TLy   ! Grid points of HPAC2D where Temp2D will start

Do
       Write(*,*) 'HPAC Data:  How many miles NORTH is NW corner of HPAC exported Output? '
       Read(*,*) NWy
       Write(*,*) 'HPAC Data:  How many miles WEST is NW corner of HPAC exported Output? '
       Read(*,*) NWx
       Write(*,600) NWy, NWx
       600 Format ("Is NE corner of HPAC data located ", i3, " miles North and ", i3, " miles
West of ground zero? (Y/N)")
       Read(*,*) Answer
       If (Answer .EQ. "Y" .OR. Answer .EQ. "y") Exit
       !NWy = 80
       !NWx = 10
End Do

TLx = -(NWx * 3)  !Where Top, Left corner is located on matrix relative to GZ
TLy = -(NWy * 3)

HPAC2D = 255.0_dp

HPAC2D(TLy:(SIZE(Temp2D, DIM=1)) - ((NWy * 3)+1), TLx:(SIZE(Temp2D, DIM=2)) - ((NWx * 3)+1)) =
Temp2D(1:SIZE(Temp2D, DIM=1),1:SIZE(Temp2D, DIM=2))

End Subroutine FillHPAC2D
!*************************************************

End Module Get_HPAC_Utility
```

Array_Utility.f90

```
Module Array_Utility

Use Kinds
Use Globals
Implicit None

Contains

Subroutine TrimArray(DASA2D, HPAC2D, Top, Bottom, Left, Right)
!*****************************************************************************************
! This subroutine trims both arrays to the bare minimum that will encompass the data of BOTH
! HPAC2D and DASA2D.  This ensures we have the same size arrays with the same ability to do
! point-to-point value comparisons.
!
!
!*****************************************************************************************

Use Kinds
Use Globals
Implicit None

Real(dp), Intent(InOut) :: DASA2D(-900:900,-900:900)! 2D array for holding dose rate values
Real(dp), Intent(InOut) :: HPAC2D(-900:900,-900:900)! 2D array for holding dose rate values
Integer,  Intent(InOut) :: Top, Bottom, Left, Right ! Overall Boundaries for final trimmed array
Character(Len = 1)      :: Direction


!Find out our limiting direction from DASA.  DASA usually cuts off data in a cardinal direction.
!We need to limit our array to the DASA distance in that direction because comparing HPAC
!dose rates beyond that distance is not valid as we do not know the dose rates beyond the data.
!If there is no cut off direction, any direction can be used.

Write(*,*) "In which cardinal direction does DASA cut off the data? (N,S,E,W, X = No Cut Off) :"
Read(*,*) Direction


If (Direction .EQ. "N" .OR. Direction .EQ. "n") Then ! Find North-most point of data comparison
       Do i = -900, 900 !If DASA is cut off to North,comparison cannot go further than cutoff
              If (MinVal(DASA2D(i, :)) .EQ. DR0) Then
                     Top = i
              Else
                     EXIT
              End If
       End Do
Else
       Do i = -900, 900       ! If DASA not cut off to North, find boundary that envelops both
                              !(DASA and HPAC)sets of data
              If (MinVal(DASA2D(i, :)) .EQ. DR0 .AND. MinVal(HPAC2D(i, :)) .EQ. DR0) Then
                     Top = i
              Else
                     EXIT
              End If
       End Do
End If

Write(*,*) "Overall Top boundary is :", TOP


If (Direction .EQ. "S" .OR. Direction .EQ. "s") Then ! Find South-most point of data comparison
  Do i = 900, -900, -1 !If DASA is cut off to South, comparison cannot go further than cutoff
       If (MinVal(DASA2D(i, :)) .EQ. DR0) Then
              Bottom = i
       Else
              EXIT
       End If
  End Do
Else
       Do i = 900, -900, -1   ! If DASA not cut off to South, find boundary that envelops both
                              !sets of data
              If (MinVal(DASA2D(i, :)) .EQ. DR0 .AND. MinVal(HPAC2D(i, :)) .EQ. DR0) Then
                     Bottom = i
              Else
                     EXIT
              End If
       End Do
End If
```

```
Write(*,*) "Overall Bottom boundary is :", Bottom

If (Direction .EQ. "W" .OR. Direction .EQ. "w") Then
        Do i = -900, 900 ! If DASA cut off to West, comparison cannot go further than cutoff
                If (MinVal(DASA2D(:, i)) .EQ. DR0) Then
                        Left = i
                Else
                        EXIT
                End If
        End Do
Else
        Do i = -900, 900 !If not cut off to West, find boundary that envelops both sets of data
                If (MinVal(DASA2D(:, i)) .EQ. DR0 .AND. MinVal(HPAC2D(:, i)) .EQ. DR0) Then
                        Left = i
                Else
                        EXIT
                End If
        End Do
End If
Write(*,*) "Overall Left boundary is :", Left



If (Direction .EQ. "E" .OR. Direction .EQ. "e") Then
        Do i = 900, -900, -1  !If DASA cut off to East, comparison cannot go further than cutoff
                If (MinVal(DASA2D(:, i)) .EQ. DR0) Then
                        Right = i
                Else
                        EXIT
                End If
        End Do
Else
        Do i = 900, -900, -1  !If DASA not cut off to East, find boundary that envelops both sets
                              !of data
                If (MinVal(DASA2D(:, i)) .EQ. DR0 .AND. MinVal(HPAC2D(:, i)) .EQ. DR0) Then
                        Right = i
                Else
                        EXIT
                End If
        End Do
End If
Write(*,*) "Overall Right boundary is :", Right

End Subroutine TrimArray
!************************************************

End Module Array_Utility
```

MOE.f90

```
Module MOEtools

Use Kinds
Use Globals
Implicit None

Contains

Subroutine CalculateStats(DASA2D, HPAC2D, Top, Bottom, Left, Right, NmbrCntrs)
!*******************************************************************************************
! This subroutine trims both arrays to the bare minimum that will encompass the data of BOTH
! HPAC2D and DASA2D.  This ensures we have the same size arrays with the same ability to do
! point-to-point value comparisons.
!*******************************************************************************************

Use Kinds
Use Globals
Implicit None

Real(dp), Intent(In)    :: DASA2D(:,:)       ! 2D array for holding dose rate values
Real(dp), Intent(In)    :: HPAC2D(:,:)       ! 2D array for holding dose rate values
Integer,  Intent(In)    :: Top, Bottom, Left, Right ! Overall Boundaries for final trimmed array
Integer,  Intent(In)    :: NmbrCntrs         ! Number of Contours to be compared
Real(dp), Allocatable   :: MOEx(:), MOEy(:)  ! X- and Y-coordinates for Measure of Effectiveness
Real(dp), Allocatable   :: NAD(:)
Real(dp)::AOB                 ! Area (represented by data points) of Observed overlay
Real(dp)::APR                 ! Area (represented by data points) of Predicted overlay
Real(dp)::AOV                 ! Area of Overlap
```

```
!Allocate Arrays
Allocate(MOEx(1:NmbrCntrs))
Allocate(MOEy(1:NmbrCntrs))
Allocate(NAD  (1:NmbrCntrs))

!Initialize Arrays
MOEx  = 0.0_dp
MOEy  = 0.0_dp
NAD   = 0.0_dp

Do i = 1, NmbrCntrs !Calculate MOE coordinates for every contour level
       AOB = 0.0_dp   !Initialize variables before each Contour Calculation
       APR = 0.0_dp
       AOV = 0.0_dp

       Do y = 1, ABS(Top-Bottom) !Calculate AOB/APR/AOV for ith contour
        Do x = 1, ABS(Left-Right)
              If (DASA2D(y,x) .LE. Contour(i)) AOB = AOB + 1._dp
                             ! Area of Observation (DASA)
              If (HPAC2D(y,x) .LE. Contour(i)) APR = APR + 1._dp
                          ! Area of Prediction  (HPAC)
              If (DASA2D(y,x) .LE. Contour(i) .AND. HPAC2D(y,x) .LE. Contour(i)) AOV=AOV+1._dp
        End Do
       End Do

       MOEx(i) = AOV/AOB !Calculate x component of MOE for ith contour
       MOEy(i) = AOV/APR !Calculate y component of MOE for ith contour
       NAD(i) = (MOEx(i) + MOEy(i) - (2.0_dp*MOEx(i)*MOEy(i))) / (MOEx(i) + MOEy(i)) !Calc NAD
End Do

Write(*,*) "Contours written from lowest level to highest level(e.g. 225=3 r/hr and 190=5 r/hr)"
Do i = 1,NmbrCntrs      ! Write out MOE coordinates and NAD value for every contour level
       Write(*,100) Contour(i), MOEx(i), MOEy(i), NAD(i)
       100 Format ("Contour : ", F7.3,5x, "MOEx: ", F7.3,5x, "MOEy: ", F7.3,5x, "NAD: ", F7.3)
End Do

Deallocate(MOEx)
Deallocate(MOEy)
Deallocate(NAD)

End Subroutine CalculateStats
!**************************************************

End Module MOEtools
```

Visualizer.f90

```
Module Visualizer

       Use Kinds
       Implicit None

       Contains
!******************************************************************************
! Created by:  Capt Rusty Williford
! Date:        19 Aug 05
! Class:               NENG 635
! Problem:     Smear Code Dose & Dose Rate Test Program
! Version:2.2 Changes: Adding Contour subroutine    Change Date:19 Aug
!       2.2.1  Refining output       22 Aug
!       2.2.2  Changes:    Added adjustable boundaries as an input NOV 05
!******************************************************************************

       Subroutine Array_Visualizer (Contour, Top, Bottom, Left, Right)
              Use AVDef
              Use DFLIB
              Implicit None

              ! Array Visualizer variables
              Integer::status
              Character(1)::key
              Real (dp), Intent(In) :: Contour(:,:)
              Real (dp), Allocatable :: Contour2(:,:)
              Integer, Intent(In):: Top, Bottom, Left, Right
              Allocate (Contour2(Top:Bottom, Left:Right))
              Contour2 = Contour
!*****************Array Visualizer Commands****************************
```

```
               !Call Starwatch to let the AView lib know we're interested in viewing M
               Call faglStartWatch(Contour2, Status)

               !View M (Brings up Array Viewer)
               Print*, "Starting Array Viewer"
               Print*, ""

               Call faglshow(Contour2,status)

               Print*, "***** Press any key to continue *****"
               key = GETCHARQQ()

               !Remove M from the watch list
               Call faglEndWatch(Contour2, status)

!*****************Array Visualizer Commands*****************************

       End Subroutine Array_Visualizer

End Module Visualizer
```

Appendix F: Research Data

George Data						
Test	Domain Size	Terrain Resolution	Contour Level	MOE X	MOE Y	NAD
George	Large	0	2	0.012	0.006	0.992
George	Large	0	0.8	0.126	0.103	0.887
George	Large	0	0.2	0.152	0.171	0.839
George	Large	0	0.08	0.163	0.160	0.839
George	Large	0	0.02	0.476	0.659	0.447
George	Large	0	0.008	0.551	0.817	0.342
George	Large	900	2	0.103	0.046	0.936
George	Large	900	0.8	0.248	0.216	0.769
George	Large	900	0.2	0.256	0.274	0.735
George	Large	900	0.08	0.264	0.213	0.764
George	Large	900	0.02	0.612	0.760	0.322
George	Large	900	0.008	0.645	0.873	0.258
George	Large	3500	2	0.140	0.063	0.913
George	Large	3500	0.8	0.261	0.217	0.763
George	Large	3500	0.2	0.246	0.305	0.728
George	Large	3500	0.08	0.250	0.233	0.759
George	Large	3500	0.02	0.684	0.767	0.277
George	Large	3500	0.008	0.718	0.863	0.216
George	Large	35000	2	0.128	0.058	0.920
George	Large	35000	0.8	0.256	0.215	0.766
George	Large	35000	0.2	0.223	0.258	0.761
George	Large	35000	0.08	0.222	0.258	0.761
George	Large	35000	0.02	0.620	0.698	0.343
George	Large	35000	0.008	0.670	0.824	0.261
George	Small	0	2	0.000	0.000	1.000
George	Small	0	0.8	0.109	0.086	0.904
George	Small	0	0.2	0.133	0.140	0.864
George	Small	0	0.08	0.142	0.115	0.873
George	Small	0	0.02	0.437	0.610	0.491
George	Small	0	0.008	0.518	0.786	0.376
George	Small	900	2	0.097	0.043	0.940
George	Small	900	0.8	0.223	0.190	0.795
George	Small	900	0.2	0.262	0.285	0.727
George	Small	900	0.08	0.285	0.252	0.733
George	Small	900	0.02	0.706	0.881	0.216
George	Small	900	0.008	0.703	0.949	0.192
George	Small	3500	2	0.087	0.040	0.945
George	Small	3500	0.8	0.244	0.210	0.774
George	Small	3500	0.2	0.244	0.210	0.774
George	Small	3500	0.08	0.229	0.217	0.777
George	Small	3500	0.02	0.660	0.776	0.287

George	Small	3500	0.008	0.695	0.883	0.222
George	Small	35000	2	0.099	0.046	0.937
George	Small	35000	0.8	0.232	0.198	0.786
George	Small	35000	0.2	0.234	0.273	0.748
George	Small	35000	0.08	0.235	0.212	0.777
George	Small	35000	0.02	0.637	0.774	0.301
George	Small	35000	0.008	0.659	0.881	0.246

Ess Data						
Test	**Domain Size**	**Terrain Resolution**	**Contour Level**	**MOE X**	**MOE Y**	**NAD**
Ess	Large	0	2	0.975	0.361	0.473
Ess	Large	0	0.8	0.964	0.465	0.373
Ess	Large	0	0.2	0.753	0.638	0.309
Ess	Large	0	0.08	0.686	0.518	0.410
Ess	Large	0	0.02	0.514	0.383	0.561
Ess	Large	0	0.008	0.459	0.355	0.600
Ess	Large	900	2	0.873	0.304	0.549
Ess	Large	900	0.8	0.897	0.450	0.401
Ess	Large	900	0.2	0.820	0.793	0.194
Ess	Large	900	0.08	0.822	0.828	0.175
Ess	Large	900	0.02	0.576	0.664	0.383
Ess	Large	900	0.008	0.556	0.675	0.390
Ess	Large	3500	2	0.816	0.286	0.577
Ess	Large	3500	0.8	0.859	0.430	0.427
Ess	Large	3500	0.2	0.782	0.742	0.239
Ess	Large	3500	0.08	0.829	0.758	0.208
Ess	Large	3500	0.02	0.585	0.612	0.402
Ess	Large	3500	0.008	0.571	0.639	0.397
Ess	Large	35000	2	0.869	0.299	0.555
Ess	Large	35000	0.8	0.898	0.436	0.413
Ess	Large	35000	0.2	0.781	0.749	0.235
Ess	Large	35000	0.08	0.832	0.782	0.194
Ess	Large	35000	0.02	0.586	0.646	0.386
Ess	Large	35000	0.008	0.534	0.583	0.443
Ess	Small	0	2	0.959	0.346	0.492
Ess	Small	0	0.8	0.958	0.456	0.382
Ess	Small	0	0.2	0.794	0.672	0.272
Ess	Small	0	0.08	0.739	0.608	0.333
Ess	Small	0	0.02	0.536	0.375	0.559
Ess	Small	0	0.008	0.517	0.370	0.569
Ess	Small	900	2	0.971	0.334	0.503
Ess	Small	900	0.8	0.959	0.451	0.387
Ess	Small	900	0.2	0.890	0.754	0.184
Ess	Small	900	0.08	0.847	0.695	0.237
Ess	Small	900	0.02	0.580	0.615	0.403

Ess	Small	900	0.008	0.516	0.576	0.456
Ess	Small	3500	2	0.943	0.321	0.521
Ess	Small	3500	0.8	0.940	0.455	0.387
Ess	Small	3500	0.2	0.843	0.792	0.183
Ess	Small	3500	0.08	0.829	0.757	0.209
Ess	Small	3500	0.02	0.577	0.564	0.430
Ess	Small	3500	0.008	0.515	0.470	0.509
Ess	Small	35000	2	0.889	0.313	0.537
Ess	Small	35000	0.8	0.916	0.474	0.375
Ess	Small	35000	0.2	0.821	0.916	0.134
Ess	Small	35000	0.08	0.838	0.677	0.251
Ess	Small	35000	0.02	0.577	0.543	0.441
Ess	Small	35000	0.008	0.498	0.530	0.487

Zucchini Data						
Test	**Domain Size**	**Terrain Resolution**	**Contour Level**	**MOE X**	**MOE Y**	**NAD**
Zucchini	Large	0	2	0.115	0.049	0.931
Zucchini	Large	0	0.8	0.115	0.065	0.917
Zucchini	Large	0	0.2	0.068	0.095	0.921
Zucchini	Large	0	0.08	0.052	0.075	0.939
Zucchini	Large	0	0.02	0.329	0.458	0.617
Zucchini	Large	0	0.008	0.449	0.659	0.466
Zucchini	Large	900	2	0.094	0.042	0.942
Zucchini	Large	900	0.8	0.080	0.047	0.941
Zucchini	Large	900	0.2	0.039	0.051	0.956
Zucchini	Large	900	0.08	0.030	0.037	0.967
Zucchini	Large	900	0.02	0.253	0.367	0.701
Zucchini	Large	900	0.008	0.335	0.538	0.587
Zucchini	Large	3500	2	0.094	0.041	0.943
Zucchini	Large	3500	0.8	0.067	0.041	0.949
Zucchini	Large	3500	0.2	0.034	0.043	0.962
Zucchini	Large	3500	0.08	0.027	0.035	0.970
Zucchini	Large	3500	0.02	0.260	0.367	0.696
Zucchini	Large	3500	0.008	0.364	0.558	0.559
Zucchini	Large	35000	2	0.092	0.041	0.943
Zucchini	Large	35000	0.8	0.069	0.038	0.951
Zucchini	Large	35000	0.2	0.034	0.045	0.961
Zucchini	Large	35000	0.08	0.027	0.035	0.970
Zucchini	Large	35000	0.02	0.271	0.399	0.677
Zucchini	Large	35000	0.008	0.353	0.581	0.561
Zucchini	Small	0	2	0.115	0.054	0.927
Zucchini	Small	0	0.8	0.111	0.058	0.924
Zucchini	Small	0	0.2	0.060	0.084	0.930
Zucchini	Small	0	0.08	0.045	0.069	0.946
Zucchini	Small	0	0.02	0.308	0.438	0.638
Zucchini	Small	0	0.008	0.432	0.643	0.483

Zucchini	Small	900	2	0.089	0.038	0.947
Zucchini	Small	900	0.8	0.057	0.034	0.957
Zucchini	Small	900	0.2	0.024	0.034	0.972
Zucchini	Small	900	0.08	0.019	0.025	0.978
Zucchini	Small	900	0.02	0.201	0.269	0.770
Zucchini	Small	900	0.008	0.298	0.435	0.646
Zucchini	Small	3500	2	0.092	0.039	0.945
Zucchini	Small	3500	0.8	0.055	0.031	0.960
Zucchini	Small	3500	0.2	0.021	0.028	0.976
Zucchini	Small	3500	0.08	0.017	0.025	0.980
Zucchini	Small	3500	0.02	0.214	0.303	0.749
Zucchini	Small	3500	0.008	0.286	0.441	0.653
Zucchini	Small	35000	2	0.097	0.044	0.940
Zucchini	Small	35000	0.8	0.067	0.038	0.952
Zucchini	Small	35000	0.2	0.033	0.040	0.964
Zucchini	Small	35000	0.08	0.025	0.035	0.971
Zucchini	Small	35000	0.02	0.241	0.345	0.716
Zucchini	Small	35000	0.008	0.298	0.482	0.632

Priscilla Data						
Test	**Domain Size**	**Terrain Resolution**	**Contour Level**	**MOE X**	**MOE Y**	**NAD**
Priscilla	Large	0	0.2	0.203	0.736	0.6818
Priscilla	Large	0	0.1	0.155	0.747	0.7433
Priscilla	Large	0	0.02	0.106	0.662	0.8173
Priscilla	Large	900	1	0.235	0.664	0.6529
Priscilla	Large	900	0.2	0.261	0.896	0.5958
Priscilla	Large	900	0.1	0.21	0.96	0.6554
Priscilla	Large	900	0.02	0.13	0.806	0.7761
Priscilla	Large	3500	1	0.242	0.707	0.6394
Priscilla	Large	3500	0.2	0.262	0.907	0.5934
Priscilla	Large	3500	0.1	0.209	0.924	0.6591
Priscilla	Large	3500	0.02	0.125	0.774	0.7848
Priscilla	Large	35000	1	0.232	0.633	0.6604
Priscilla	Large	35000	0.2	0.247	0.871	0.6151
Priscilla	Large	35000	0.1	0.209	0.994	0.6546
Priscilla	Large	35000	0.02	0.126	0.787	0.7828
Priscilla	Small	0	1	0.274	0.783	0.5941
Priscilla	Small	0	0.2	0.189	0.655	0.7066
Priscilla	Small	0	0.1	0.135	0.654	0.7762
Priscilla	Small	0	0.02	0.07	0.432	0.8795
Priscilla	Small	900	1	0.202	0.578	0.7006
Priscilla	Small	900	0.2	0.246	0.85	0.6184
Priscilla	Small	900	0.1	0.202	0.938	0.6676
Priscilla	Small	900	0.02	0.109	0.692	0.8117
Priscilla	Small	3500	1	0.228	0.612	0.6678

Priscilla	Small	3500	0.2	0.253	0.874	0.6076
Priscilla	Small	3500	0.1	0.201	0.948	0.6683
Priscilla	Small	3500	0.02	0.112	0.713	0.8064
Priscilla	Small	35000	1	0.221	0.634	0.6722
Priscilla	Small	35000	0.2	0.259	0.879	0.5999
Priscilla	Small	35000	0.1	0.204	0.947	0.6643
Priscilla	Small	35000	0.02	0.118	0.679	0.7989

Smoky Data						
Test	**Domain Size**	**Terrain Resolution**	**Contour Level**	**MOE X**	**MOE Y**	**NAD**
Smoky	Large	0	10	0.003	0.017	0.995
Smoky	Large	0	2	0.000	0.000	1.000
Smoky	Large	0	1	0.000	0.001	1.000
Smoky	Large	0	0.2	0.000	0.002	1.000
Smoky	Large	0	0.1	0.001	0.013	0.998
Smoky	Large	0	0.02	0.000	0.000	1.000
Smoky	Large	900	20	0.003	0.006	0.996
Smoky	Large	900	10	0.002	0.013	0.997
Smoky	Large	900	2	0.000	0.000	1.000
Smoky	Large	900	1	0.000	0.001	1.000
Smoky	Large	900	0.2	0.000	0.002	1.000
Smoky	Large	900	0.1	0.001	0.013	0.998
Smoky	Large	900	0.02	0.000	0.000	1.000
Smoky	Large	3500	20	0.003	0.006	0.996
Smoky	Large	3500	10	0.002	0.013	0.997
Smoky	Large	3500	2	0.000	0.000	1.000
Smoky	Large	3500	1	0.000	0.001	1.000
Smoky	Large	3500	0.2	0.000	0.002	1.000
Smoky	Large	3500	0.1	0.001	0.013	0.998
Smoky	Large	3500	0.02	0.000	0.000	1.000
Smoky	Large	35000	20	0.003	0.007	0.996
Smoky	Large	35000	10	0.002	0.013	0.997
Smoky	Large	35000	2	0.000	0.000	1.000
Smoky	Large	35000	1	0.000	0.001	1.000
Smoky	Large	35000	0.2	0.000	0.002	1.000
Smoky	Large	35000	0.1	0.001	0.012	0.998
Smoky	Large	35000	0.02	0.000	0.000	1.000
Smoky	Small	0	20	0.004	0.009	0.995
Smoky	Small	0	10	0.002	0.016	0.996
Smoky	Small	0	2	0.000	0.000	1.000
Smoky	Small	0	1	0.000	0.001	1.000
Smoky	Small	0	0.2	0.000	0.002	1.000
Smoky	Small	0	0.1	0.001	0.012	0.998
Smoky	Small	0	0.02	0.000	0.000	1.000
Smoky	Small	900	20	0.003	0.007	0.996

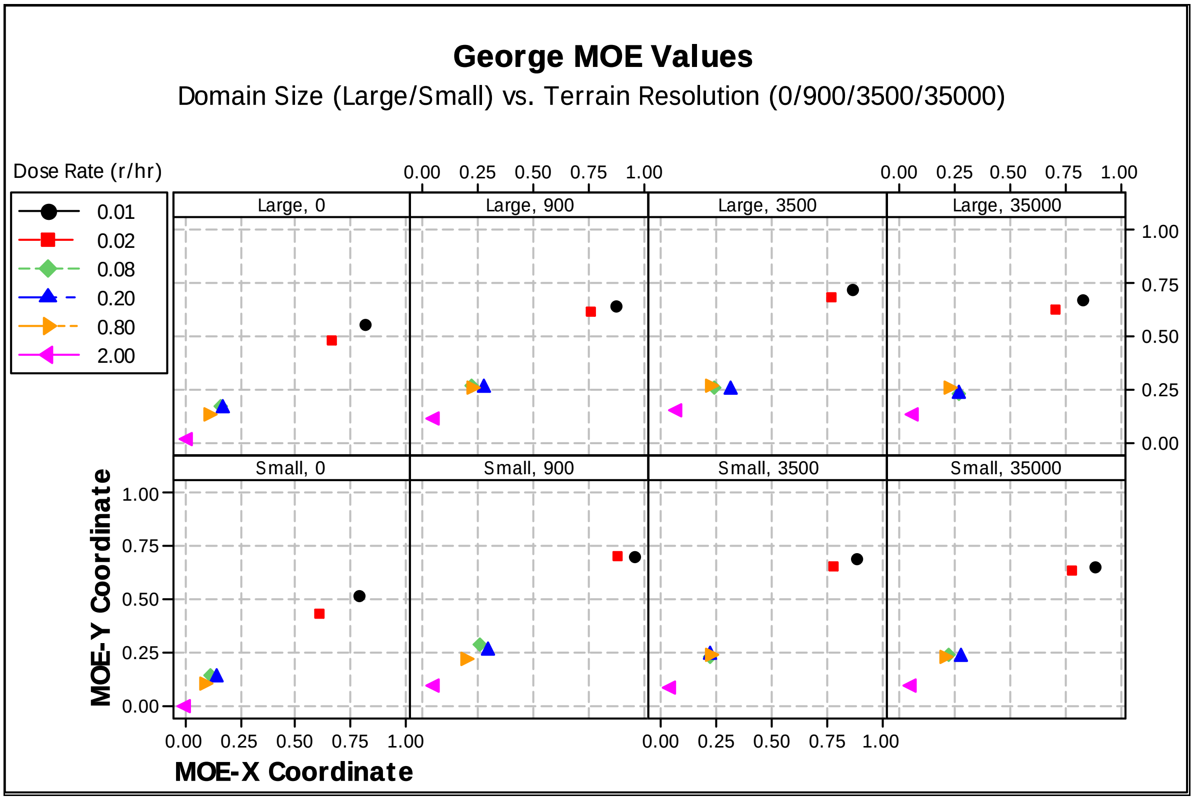
Smoky	Small	900	10	0.002	0.015	0.997
Smoky	Small	900	2	0.000	0.000	1.000
Smoky	Small	900	1	0.000	0.001	1.000
Smoky	Small	900	0.2	0.000	0.002	1.000
Smoky	Small	900	0.1	0.001	0.013	0.998
Smoky	Small	900	0.02	0.000	0.000	1.000
Smoky	Small	3500	20	0.004	0.008	0.995
Smoky	Small	3500	10	0.002	0.015	0.997
Smoky	Small	3500	2	0.000	0.000	1.000
Smoky	Small	3500	1	0.000	0.001	1.000
Smoky	Small	3500	0.2	0.000	0.002	1.000
Smoky	Small	3500	0.1	0.001	0.012	0.998
Smoky	Small	3500	0.02	0.000	0.000	1.000
Smoky	Small	35000	20	0.003	0.005	0.996
Smoky	Small	35000	10	0.002	0.013	0.997
Smoky	Small	35000	2	0.000	0.000	1.000
Smoky	Small	35000	1	0.000	0.001	1.000
Smoky	Small	35000	0.2	0.000	0.002	1.000
Smoky	Small	35000	0.1	0.001	0.011	0.998
Smoky	Small	35000	0.02	0.000	0.000	1.000

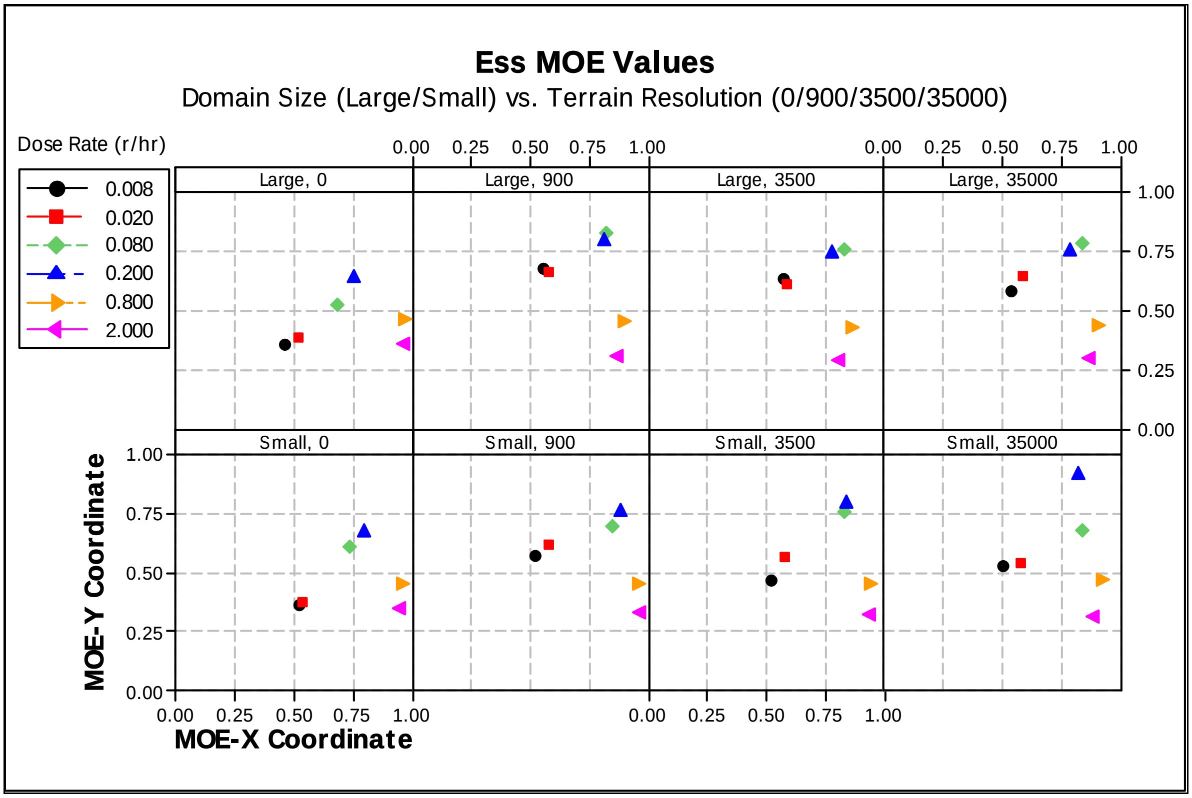
Johnie Boy Data

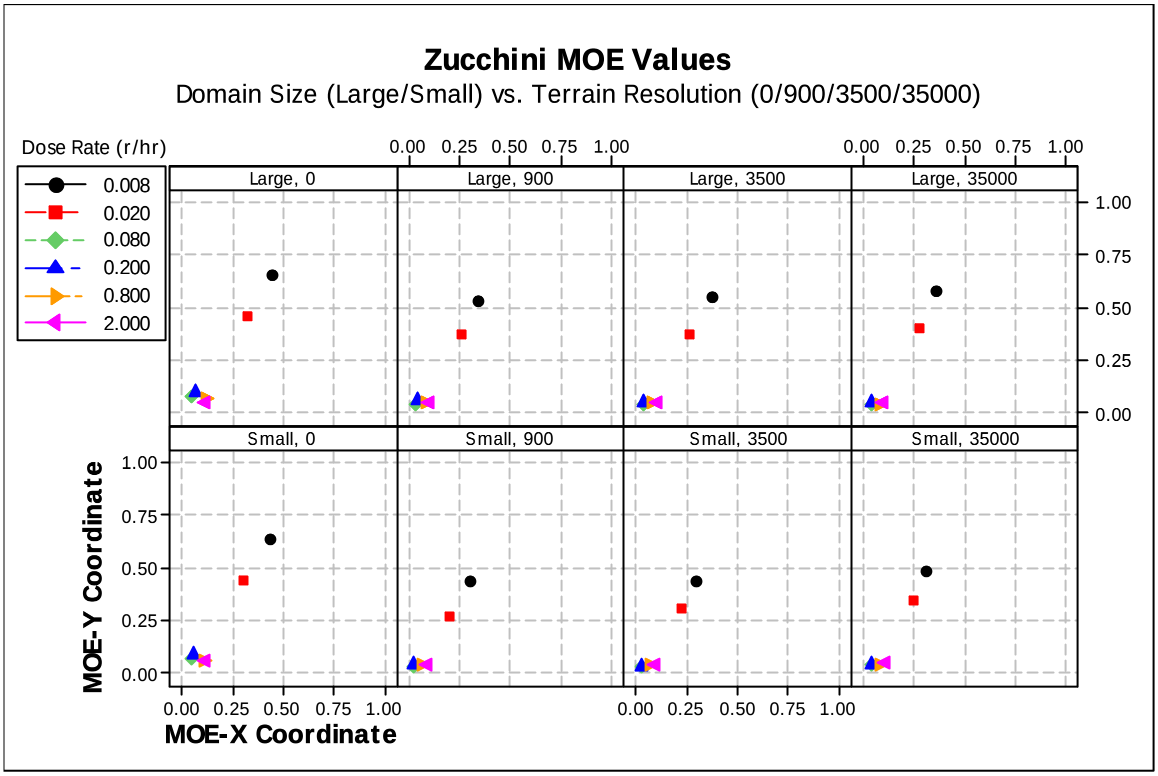
Test	Domain Size	Terrain Resolution	Contour Level	MOE X	MOE Y	NAD
Johnie Boy	Large	0	1	0.077	0.037	0.950
Johnie Boy	Large	0	0.5	0.048	0.031	0.962
Johnie Boy	Large	0	0.1	0.028	0.011	0.984
Johnie Boy	Large	0	0.05	0.024	0.008	0.988
Johnie Boy	Large	0	0.01	0.024	0.010	0.986
Johnie Boy	Large	900	10	0.188	0.108	0.863
Johnie Boy	Large	900	1	0.062	0.026	0.963
Johnie Boy	Large	900	0.5	0.046	0.027	0.966
Johnie Boy	Large	900	0.1	0.027	0.010	0.985
Johnie Boy	Large	900	0.05	0.024	0.009	0.987
Johnie Boy	Large	900	0.01	0.050	0.024	0.968
Johnie Boy	Large	3500	10	0.188	0.108	0.863
Johnie Boy	Large	3500	1	0.062	0.023	0.966
Johnie Boy	Large	3500	0.5	0.046	0.024	0.969
Johnie Boy	Large	3500	0.1	0.027	0.011	0.984
Johnie Boy	Large	3500	0.05	0.027	0.014	0.982
Johnie Boy	Large	3500	0.01	0.062	0.029	0.961
Johnie Boy	Large	35000	10	0.188	0.103	0.867
Johnie Boy	Large	35000	1	0.062	0.024	0.965
Johnie Boy	Large	35000	0.5	0.047	0.026	0.967
Johnie Boy	Large	35000	0.1	0.029	0.013	0.982
Johnie Boy	Large	35000	0.05	0.030	0.011	0.984
Johnie Boy	Large	35000	0.01	0.073	0.034	0.954

Johnie Boy	Small	0	10	0.229	0.141	0.826
Johnie Boy	Small	0	1	0.083	0.038	0.948
Johnie Boy	Small	0	0.5	0.053	0.034	0.959
Johnie Boy	Small	0	0.1	0.032	0.013	0.982
Johnie Boy	Small	0	0.05	0.028	0.009	0.986
Johnie Boy	Small	0	0.01	0.040	0.017	0.976
Johnie Boy	Small	900	10	0.146	0.082	0.895
Johnie Boy	Small	900	1	0.062	0.025	0.964
Johnie Boy	Small	900	0.5	0.040	0.023	0.971
Johnie Boy	Small	900	0.1	0.025	0.010	0.986
Johnie Boy	Small	900	0.05	0.026	0.009	0.987
Johnie Boy	Small	900	0.01	0.060	0.026	0.964
Johnie Boy	Small	3500	10	0.146	0.083	0.894
Johnie Boy	Small	3500	1	0.062	0.025	0.964
Johnie Boy	Small	3500	0.5	0.039	0.024	0.970
Johnie Boy	Small	3500	0.1	0.024	0.010	0.986
Johnie Boy	Small	3500	0.05	0.024	0.009	0.987
Johnie Boy	Small	3500	0.01	0.063	0.028	0.961
Johnie Boy	Small	35000	10	0.250	0.135	0.825
Johnie Boy	Small	35000	1	0.092	0.037	0.947
Johnie Boy	Small	35000	0.5	0.064	0.037	0.953
Johnie Boy	Small	35000	0.1	0.042	0.020	0.973
Johnie Boy	Small	35000	0.05	0.046	0.017	0.975
Johnie Boy	Small	35000	0.01	0.085	0.037	0.948

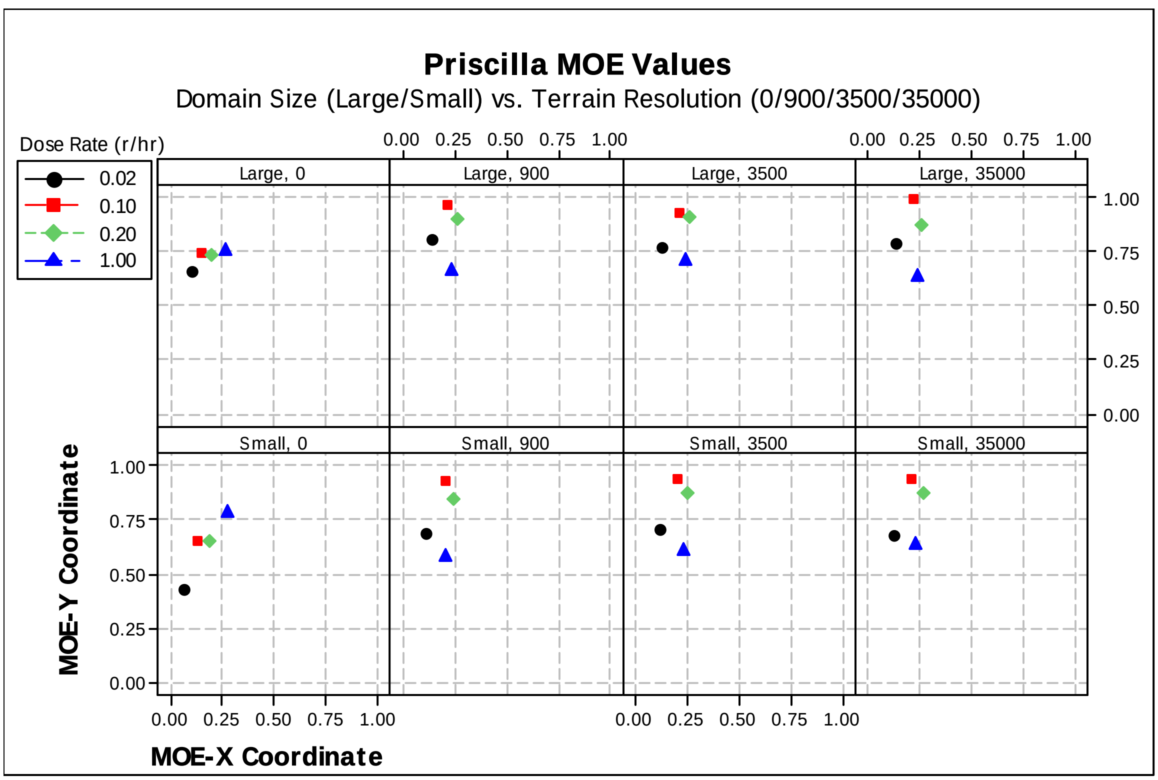
Appendix G: MOE Plots



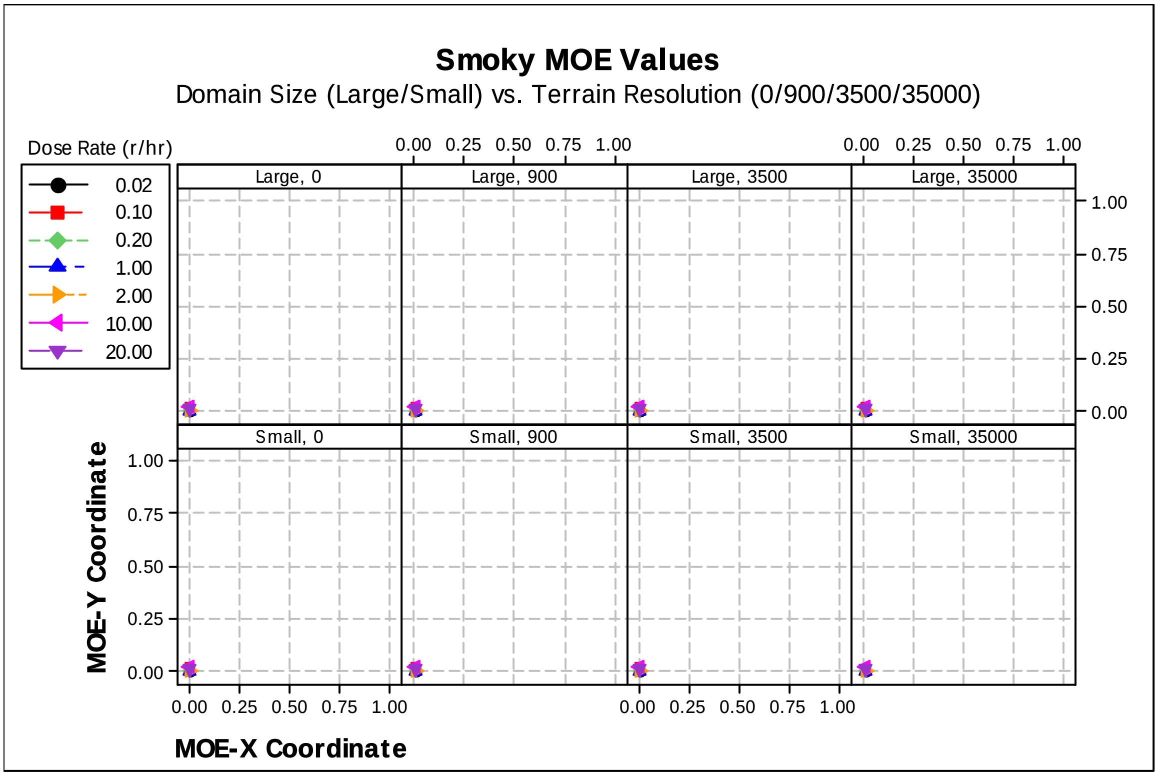




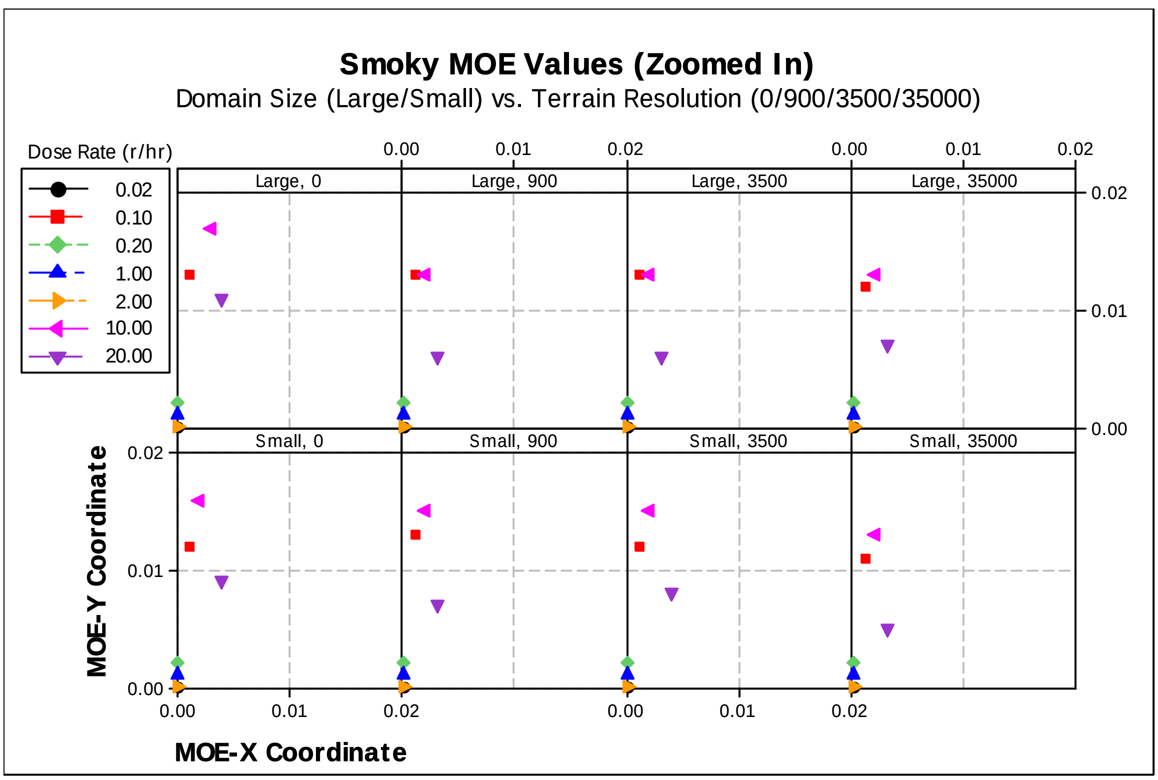
Zucchini MOE Values
Domain Size (Large/Small) vs. Terrain Resolution (0/900/3500/35000)
Dose Rate (r/hr)
0.008
0.020
0.080
0.200
0.800
2.000
Large, 0
Large, 900
Large, 3500
Large, 35000
Small, 0
Small, 900
Small, 3500
Small, 35000
0.00 0.25 0.50 0.75 1.00
MOE-Y Coordinate
MOE-X Coordinate



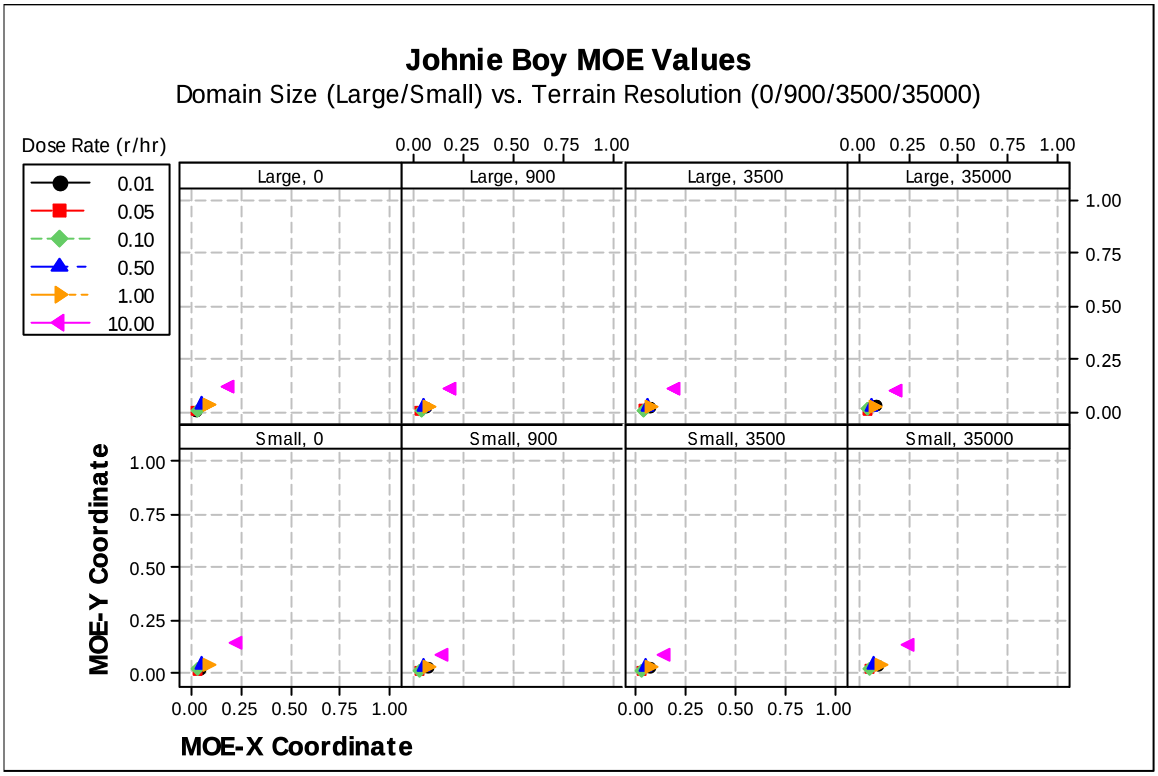
Priscilla MOE Values
Domain Size (Large/Small) vs. Terrain Resolution (0/900/3500/35000)
Dose Rate (r/hr)
0.02
0.10
0.20
1.00
Large, 0
Large, 900
Large, 3500
Large, 35000
Small, 0
Small, 900
Small, 3500
Small, 35000
0.00 0.25 0.50 0.75 1.00
MOE-Y Coordinate
MOE-X Coordinate

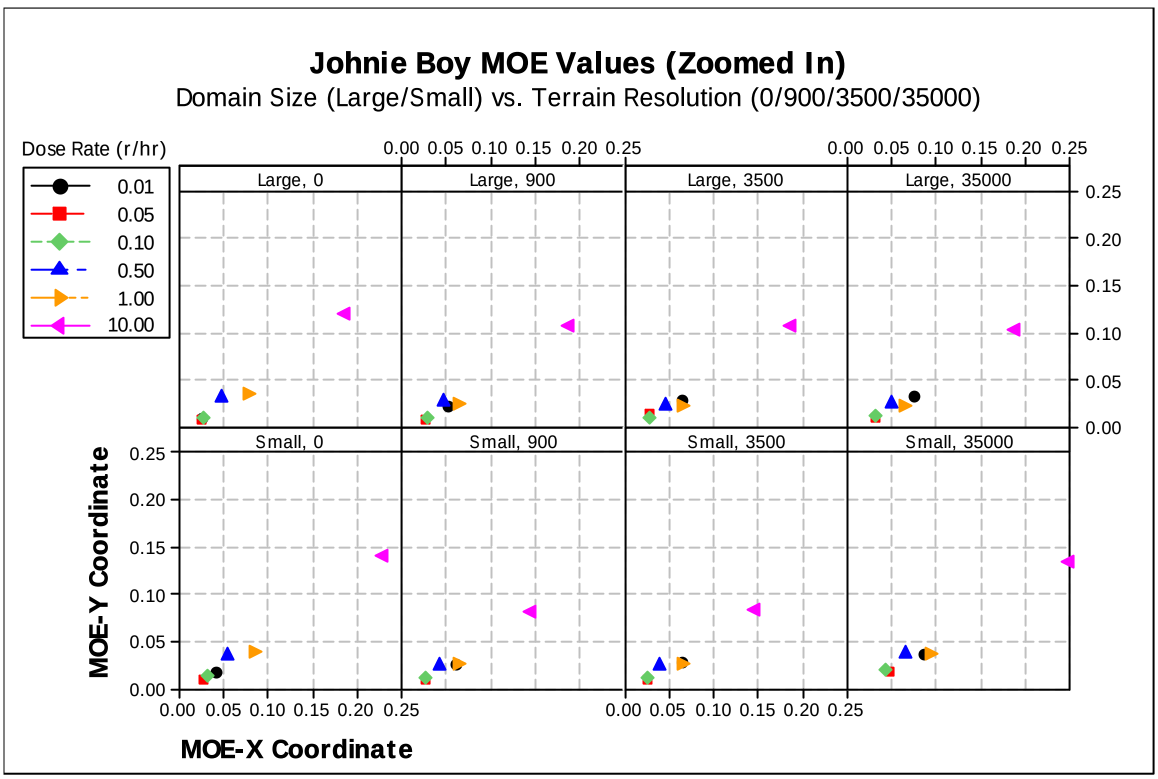


Smoky MOE Values
Domain Size (Large/Small) vs. Terrain Resolution (0/900/3500/35000)
Dose Rate (r/hr)
0.02
0.10
0.20
1.00
2.00
10.00
20.00
Large, 0
Large, 900
Large, 3500
Large, 35000
Small, 0
Small, 900
Small, 3500
Small, 35000
0.00 0.25 0.50 0.75 1.00
MOE-Y Coordinate
MOE-X Coordinate



Smoky MOE Values (Zoomed In)
Domain Size (Large/Small) vs. Terrain Resolution (0/900/3500/35000)
Dose Rate (r/hr)
0.02
0.10
0.20
1.00
2.00
10.00
20.00
Large, 0
Large, 900
Large, 3500
Large, 35000
Small, 0
Small, 900
Small, 3500
Small, 35000
0.00 0.01 0.02
MOE-Y Coordinate
MOE-X Coordinate


Johnie Boy MOE Values
Domain Size (Large/Small) vs. Terrain Resolution (0/900/3500/35000)
Dose Rate (r/hr)
0.01
0.05
0.10
0.50
1.00
10.00
Large, 0
Large, 900
Large, 3500
Large, 35000
Small, 0
Small, 900
Small, 3500
Small, 35000
0.00 0.25 0.50 0.75 1.00
MOE-Y Coordinate
MOE-X Coordinate


Johnie Boy MOE Values (Zoomed In)
Domain Size (Large/Small) vs. Terrain Resolution (0/900/3500/35000)
Dose Rate (r/hr)
0.01
0.05
0.10
0.50
1.00
10.00
Large, 0
Large, 900
Large, 3500
Large, 35000
Small, 0
Small, 900
Small, 3500
Small, 35000
0.00 0.05 0.10 0.15 0.20 0.25
MOE-Y Coordinate
MOE-X Coordinate

Bibliography

1. Department of Energy. *United States Nuclear Tests: July 1945 through September 1992*. Santa Monica CA: The RAND Corporation, June 1969 (RM-5888-PR).

2. DASA 1251-1-EX, *Compilation of Local Fallout Data from Test Detonations 1945-1962 Extracted from DASA 1251. Volume I – Continental U.S. Tests*. Defense Nuclear Agency, Washington D.C., 1979

3. Chancellor, Richard W. *A Comparison of Hazard Prediction and Assessment Capability (HPAC) Software Dose-Rate Contour Plots to a Sample of Local Fallout Data from Test Detonations in the Continental United States*, 1945-1962. Air Force Institute of Technology, Wright-Patterson AFB, March 2005

4. Warner, Steve and others. "User-Oriented Two-Dimensional Measure of Effectiveness for the Evaluation of Transport and Dispersion Models," Journal of Applied Meteorology, 43: 58-73 (January 2004)

5. Chancellor, Richard W. *Comparison of Historical Nuclear Fallout Data to HPAC Using Reanalysis Weather Data.* Air Force Institute of Technology, Wright-Patterson AFB, May 2005

6. Glasstone, Samuel and Dolan, Philip J. *The Effects of Nuclear Weapons*. Third Edition, Department of Defense and Department of Energy, 1977

7. Bridgman, Charles J. *Introduction to the Physics of Nuclear Weapons Effects*. Defense Threat Reduction Agency, 8725 John J. Kingman Road, Fort Belvoir VA 22060-6201, 2001.

8 *Google Earth*. Version 3.0.0739, Computer Software, Google Inc., 2003. http://earth.google.com/

9. Defense Thread Reduction Agency. *HPAC 4.04 Users Manual*, Alexandria VA, April 2004.

10. Defense Thread Reduction Agency. *HPAC 4.04 Basic Course*, Alexandria VA, http://www.dtratraining.net

11. DTRA Technical Reachback Team. "HPAC Terrain Data and Version Summary." Electronic Message. 074600L, 3 November 2005.

12. Kalnay, Eugenia and others. "The NCEP/NCAR 40-Year Reanalysis Project," *Bulletin of the American Meteorological Society*. 77: 437-431 (3 March 1996).

13. Defense Threat Reduction Agency. *Hazard Prediction and Assessment Capability (HPAC) User Guide Version 4.0.3*, Alexandria VA, 9 May 2003.

14. National Oceanic and Atmospheric Administration. “NOAA Operational Model Archive Distribution System.” Camp Springs MD: October 2005. http://nomad3.ncep.noaa.gov/

15. Compaq Visual FORTRAN, Professional Edition. Version 6.6.0, CD-ROM, Computer Software. Hewlett-Packard USA, Houston TX, 2000

16. Wesley Ebisuzaki. “Reading GRIB files,” Unpublished document. N. pag. http://www.cpc.ncep.noaa.gov/products/wesley/reading_grib.html

17. *Canvas*. Version 9.0.4 Build 820, CD-ROM, Computer Software. ACD Systems of America, Miami FL, 2003.

18. Turner, James E. *Atoms, Radiation, and Radiation Protection* (2nd Edition). New York: John Wiley & Sons, Inc., 1995.

19. Commission on Engineering and Technical Systems National Research Council. “Film Badge Dosimetry in Atmospheric Tests.” Washington D.C.: National Academy Press, 1989.

20. *Minitab*. Release 14.20, Computer Software. Minitab Inc., State College PA, 2005.

Vita

Major Kevin D. Pace graduated from Bedford North Lawrence High School in Bedford, Indiana. He entered undergraduated studies at Indiana University located in Bloomington, Indiana where he graduated with a Bachelor of Science degree in Education. He was commissioned into the US Army Chemical Corps through the IU's Screaming Bison ROTC battalion.

After his initial schooling at Fort McClellan, Alabama, he served as a Chemical officer in an Apache Helicopter battalion in the 101st Airborne Divisions (Air Assault). In 1996, Major Pace completed his branch detail in the Chemical Corps and was assigned as a Quartermaster platoon leader in the 801st Main Support Battalion. Upon his tour completion at Fort Campbell, KY, he attended the Officer Advance Course at Fort Lee, VA. He then served as a company commander at Camp Foster located in Okinawa, Japan. He was then assigned to Fort Knox, Kentucky in 1998 where he served as the company commander of Bravo Company, US Army Armor Center followed by an assignment in the 3-337 Training Support Battalion where he initially served as an Observer/Controller/Trainer and finally as the Assistant Battalion Operations Officer. He entered the Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH. Upon graduation, he will serve as an Army Nuclear and Nonproliferation officer at US Northern Command, Peterson AFB located in Colorado Springs.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
23-03-2006	**Master's Thesis**	Oct 2005 − Mar 2006

4. TITLE AND SUBTITLE	
Terrain and Spatial Effects on Hazard Prediction and Assessment Capability (HPAC) Software Dose-Rate Contour Plot Predictions as Compared to a Sample of Local Fallout Data from Test Detonations in the Continental United States, 1945-1962	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Pace, Kevin, D, MAJOR, USA	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GNE/ENP/06-06
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Hazard Prediction and Capability (HPAC) Software is validated by comparing modeled predictions to historical test data. Reanalysis weather data is acquired and reformatted for use in HPAC. Simulations are made using various amounts of weather data by use of a spatial domain. Simulations are also varied by levels of terrain resolution. The predicted output of the software is numerically compared to historical test data. The result of this research culminated in the knowledge that HPAC prediction accuracy is improved by using terrain resolutions beyond the flat earth assumption. Furthermore, this research establishes that domain size variation produces no significant advantage as to the accuracy of the prediction.

15. SUBJECT TERMS

Fallout, Radioactive Fallout, HPAC, Nuclear Test Detonations, Measure of Effectiveness, Normalized Absolute Difference, Areal Comparison, Downwind Hazard, Dose-Rate Contours

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Lt Col Steven T. Fiorino, AFIT/ENP
REPORT	**ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	159	(937) 255-3636, ext 4506

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18